THE MESONH USER'S GUIDE

MASDEV4.9.1 version and SURFEX 7.1 version

February 13, 2012

Contents

Chapter 1

Introduction

MESONH is an atmospheric simulation model which can be run in very different conditions. Its capabilities are completely described in the scientific documentation of the model, where the model equations and the whole physics are given.

This book gives the necessary information to perform numerical experiments, using the MESONH atmospheric system. It will help the user to install MESO-NH and to prepare a numerical experiment made by this model.

From a technical point of view, no special knowledge are required to perform a simulation. It is useful to specify the main characteristics of the model:

- The sources are written in **standard Fortran 90** (Metcalf and Reid 1993)
- The source file management is perform by **CVS**
- The generalized use of **dynamic memory allocation** avoids repeated compilation of the source and the free parameters are set by the user through namelist files

To realize an experiment with MESONH, the user will made a sequence of elementary steps. This sequence will differ according to the type of the simulation : ideal or real, with grid-nesting or not. The different steps are :

1. **PREP_PGD**: this program computes the physiographic data file. At this step, you choose the projection, horizontal resolution and domain. The PGD file contains all the physiographic data necessary to run the MESO-NH model with interactive surface schemes for vegetation and town. PREP_PGD is presented in chapter 4.
2. **PREP_NEST_PGD** : this MESO-NH program checks all the PGD files to impose conformity of orography between them. This program is only used with grid-nesting simulations. (see chapter 4)

3. **ZOOM_PGD** : this program allows to zoom a PGD file on the part of interest to make an inner domain at the same resolution. (see chapter 4)

4. **PREP_IDEAL_CASE** : this MESO-NH program prepares an initial MESONH file, that contains all the parameters and fields necessary for the execution of the model. (grid parameters, initial fields and geophysical fields). It is presented in chapter 5.

5. **extractecmwf**[1] or **extractarpege**[2] : it extracts the surface and altitude fields for a specific date, respectively for ECMWF archive (ECMWF forecast model) or METEO-FRANCE operational archive (ARPEGE, AROME and ALADIN models). In both cases, the fields are written in a GRIB format file, on the gaussian grid. The extraction must be done separately for each date and time (for the initial file and each of the coupling file). The Unix procedures used for these extractions are not presented in this book.

6. **PREP_REAL_CASE** : this MESO-NH program performs the change of orography and vertical grid by interpolating horizontally and vertically for a GRIB file or only vertically for a MESO-NH file. The MESO-NH output file will be used either for the beginning of the simulation or for coupling. It is presented in chapter 6.

7. **SPAWNING** : this program performs the horizontal interpolations from a MESO-NH file into another MESO-NH file, with a finer resolution and smaller domain. (see chapter 7)

8. **MESONH** : it is the temporal integration of the model. This step is developped in details in chapter 9.

9. **DIAG** : this program performs diagnostic variables after the simulation. To have more information on this program see chapter 10.

10. **SPECTRE** : this program, presented in chapter 11, performs a particular diagnostic : energy spectra.

The figures 1.1 and 1.2 present the sequence of the steps for an ideal simulation and a real simulation in mono-model case and with grid-nesting respectively. Detailed sequences for grid-nesting simulation are described in annexe B.

[1]A user account is necessary on the station ecgate of ECMWF to run **extractecmwf**
[2]A user account at METEO FRANCE is necessary to run **extractarpege**

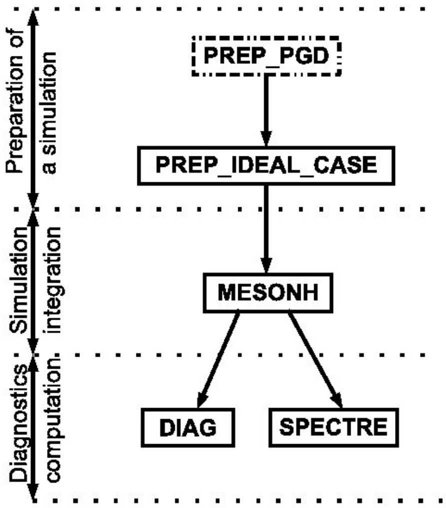
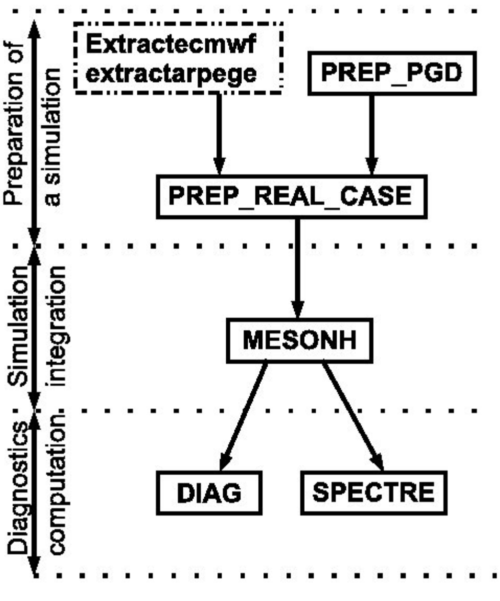
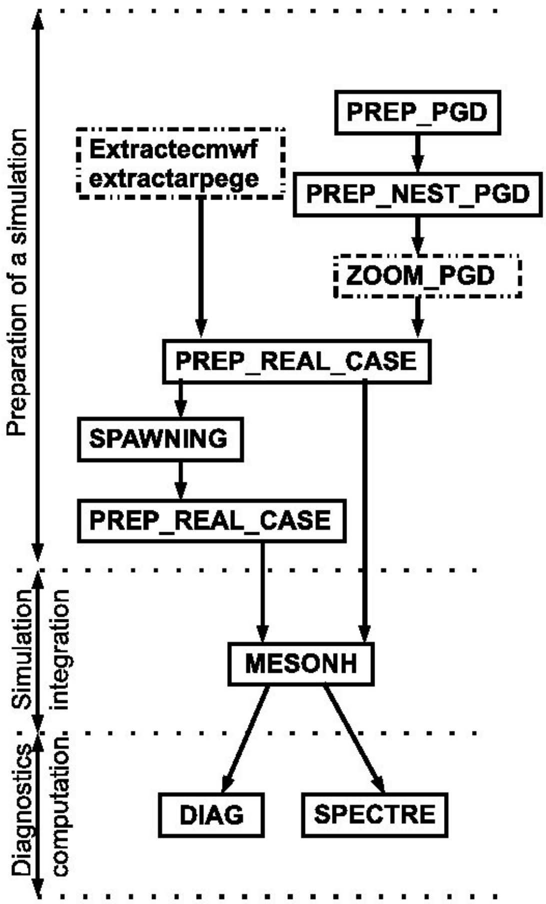


Figure 1.1 : General algorithm for an idealized numerical experiment



mono-model



with grid-nesting

Figure 1.2 : General algorithm for a real numerical experiment

First, in chapter 2, we wil present how to install MESONH and how to compile personal sources. Then in chapter 3, the MESONH files (namelists and FM file) are described in details. In chapter 4 to 11, we will describe every elementary step of MESONH. In Annexe A, we will list all the variables present in a FM-file, in annexe B, we will describe sequences for grid-nesting simulation ; in annexe C, we present in details the LES diagnostics. Finally, in annexe D you will find a description of the MESO-NH grid.

Chapter 2

Installation of MESONH

This chapter is a part of the file A-INSTALL present in the MESO-NH package.

2.1 Downloading MESONH

MESONH sources and executables are developed and maintained with the CVS tools (http://www.cvshome.org/). There are two ways to download the new package of MESONH containing :

- sources
- makefiles
- precompile exe
- graphic tools
- basic examples

The first way is for USER/DEVELOPER of MESONH via the use of the CVS tools and an access via anonymous connection with ”ssh” to the CVS REPOSITORY of the MESONH package.
The second way is for VERY BASIC USER OF MESONH via a download of a ”tar ball” in the WEB site of MESONH.

2.1.1 Downloading MESONH via CVS ANONYMOUS

Download the ssh key file ”anoncvs.key” for anonymous connection

With your preferred web browser go to the MESONH WEB SITE

```
http://mesonh.aero.obs-mip.fr/mesonh
    ---> Download
          ---> CVS PACKAGE MESONH
```

or directly

```
http://mesonh.aero.obs-mip.fr/cgi-bin/mesonh_interne/viewcvs.cgi/MNH-VX-Y-Z
```

In the field "Show files using tag:", select "PACK-MNH-V4-9-1".
Download the file `anoncvs.key` by :

1. a "left-click" in the "Rev." column
2. or a "right-click" on "download". Save link to disk (Warning :: don't copy the file content with mouse copy/past because it contains binary-encoded information !!!)

Then copy this file in your `${HOME}/.ssh/` directory
AND VERY IMPORTANT, modify the "read/write" permission with :

```
chmod 600 anoncvs.key
```

Download the config file `config.anoncvs_www`

For METEO-FRANCE & IDRIS & CINES & Laboratoire d'Aerologie computers (IP address filter)
Download the file `config.anoncvs_www`
Then concatened the file content with your `${HOME}/.ssh/config` file (this will define the computer alias `mesonh_anoncvs_www` for future ssh anonymous connections)

```
cd ${HOME}/.ssh
cat config.anoncvs_www >> config
```

For other computers use special `config.anoncvs_www`
if the computer, from which you download the MESONH sources, is external to METEO-FRANCE & IDRIS & CINES & Laboratoire d'Aerologie Laboratoire download this file :
`config.anoncvs_www_ext`

```
cd ${HOME}/.ssh
cat config.anoncvs_www_ext >> config
```

Or for ECMWF computer (c1a) download this config file (to bypass the gateway filter) :
`config.anoncvs_www_ecmwf`

```
cd ${HOME}/.ssh
cat config.anoncvs_www_ecmwf >> config
```

Setting CVS variables

set the `CVS_RSH` and `CVSROOT` (in your `.profile` or `.bashrc` file) like this :

```
export CVS_RSH=ssh
export CVSROOT=:ext:mesonh_anoncvs_www:/home/cvsroot
```

Checking out the "MESONH PACKAGE"

Now, from your `$HOME` directory for example extract the version "PACK-MNH-V4-9-1" of the directory "MNH-VX-Y-Z" from the cvs repository :

```
cd ~
cvs co -r PACK-MNH-V4-9-1 -d MNH-V4-9-1 MNH-VX-Y-Z
```

WARNING : don't use a sub-directory with dot "." in the name (you could have some trouble when compiling mesonh)

This will create in your `$HOME` a directory "MNH-V4-9-1" which contains the last revision named "PACK-MNH-V4-9-1" of the MESONH PACKAGE.
The advantage of this way of downloading the package is that in the future you could check/update quickly differences with the new version of the package without having to download entirely the full package.
Suppose that a new version for example "PACK-MNH-V4-9-1" is announced ... To see the differences with your working copy do :

```
 cd ~/MNH-V4-9-1
 cvs diff -r PACK-MNH-V4-9-1
```

And to upgrade your working copy :

```
 cd ~/MNH-V4-9-1
cvs update -r PACK-MNH-V4-9-1 -d -P
```

Beginning with the "PACK-MNH-V4-7-1" release at any time you could also check for "upto-date" changes in the CVS "branch" dedicated to the MNH49 version before the official release of the "bugN+1" bugfix

2.1.2 Downloading MESONH via the MESONH web site

With your preferred web browser go to the MESONH web site

```
  http://mesonh.aero.obs-mip.fr/mesonh
      ---> Download
            ---> CVS PACKAGE MESONH
```

or directly

```
http://mesonh.aero.obs-mip.fr/cgi-bin/mesonh_interne/viewcvs.cgi/MNH-VX-Y-Z
```

In the field "Show files using tag:", select "PACK-MNH-V4-9-1" and then download the file "MNH-VX-Y-Z.tar.gz" with the link `" Download tarball "`
Then untar the file "MNH-VX-Y-Z.tar.gz" where you want to, in your home directory for example:
`cd ~`
`tar xvfz MNH-VX-Y-Z.tar.gz`
As the directory did not reflect the last version name move it to the right one

```
mv MNH-VX-Y-Z MNH-V4-9-1
```

2.2 Installing/configuring the MESONH package

For the installation process, you could now use the `configure` script like this :
`cd ~/MNH-V4-9-1/src`
`configure`
This will create a configuration file `profile_mesonh` with an extension reflecting the different "choices" made automatically to match the computer on which you want to install MESONH and then source/load the new generate file :

```
. ../conf/profile_mesonh
```

WARNING : on GENCI, ECMWF and METEO-FRANCE computers, the configure is tuned to identify the computer on which the command is used so the good compiler, mpi & cdf library , etc ... is automatically chosen
This is not the case in your "own" personal Linux computer ...It is up to you to set the ARCH variable correctly ARCH= Fortran compiler to use, and all the other environnement variables . By default :

- the compiler is chosen to be "g95" : `ARCH=LXg95`
- the mpi library to be the MPIVIDE :`VER_MPI=MPIVIDE` (empty mpi library coming with MESONH package = no parallel run possible)
- the level compiler optimization : `OPTLEVEL=DEBUG` (for development purpose, fast compilation, debugging)

So if needed, you could also change the default FLAG compiler/mpi/optlevel like this, for example :

```
export ARCH=LXifort         # Use Intel "ifort" compiler on LX=linux Plateform
export VER_MPI=OMPI12X      # Use OPEN-MPI version 1.2.X
export OPTLEVEL=O2          # Compile in O2 , 4 time faster then DEBUG,
                              but least error check
```

```
./configure
```

and then source/load the new generate file

```
. ../conf/profile_mesonh.LXifort.MNH-V4-9-1.OMPI12X.O2
```

Note :

Options specific to compile/architecture, like 'OPTLEVEL' are defined inside the "Rules.${ARCH}.mk".
Options specific to library like "mpi"="VER_MPI" or "cdf"="VER_CDF" are defined inside "Makefile.MESONH.mk"
If needed for adaptation to your requirement, look inside the files and changes options according to your needs .
On PC-Linux, look in the "MesonhTEAM Wiki" to know how to compile the library MPI = OPEN-MPI with MESONH

```
http://mesonh.aero.obs-mip.fr/teamwiki/MesonhTEAMFAQ/PC\_Linux
     --> Compilation of OPEN-MPI
```

2.3 Compiling the MESONH package

2.3.1 On PC Linux

Go to the directory "src"

```
cd ~/MNH-V4-9-1/src
```

If you have not already configured your MESONH environment either manually in your interactive session or automatically through your .profile (or .bashrc), do:

```
. ../conf/profile_mesonh
```

(use the configure file corresponding to your needs (see section 2.2))

Run the compilation by :

```
(g)make
```

The compilation will take about 20 minutes on modern PC-Linux ... If you have a multi-processor machine you can speedup the compilation, for example, on two processors with:

`(g)make -j 2`

The object files "*.o" and main executables of the "MESONH PACKAGE" : MESONH, PREP_IDEAL_CASE, PREP_REAL_CASE... are compiled in one step and created in the directory :

`dir_obj-$(ARCH).../MASTER`

The exact name of this `dir_obj...` depends on the different environnement variables set by the `profile_mesonh....` which you have loaded before the compilation.
This allows to use different configurations in the same direction by loading different `profile_mesonh..` files to compile in the same installation directory.

To install the new compiled program in the `$SRC_MESONH/exe` directory, after compilation, just run :

`make installmaster`

The executable with their full name, including, $ARCH , compiler and MPI , level of optimization, will be linked in the `../exe` directory Note : the `make installmaster` need to be done only once by *version*. If you only change/add source, you just have to do `make`

2.3.2 On GENCI, ECMWF or METEO-FRANCE platforms

Due to limitation in time and memory on interactive connection, the MESONH PACKAGE must be compiled in batch mode with the different `job_make_mesonh*` files

On BABEL (IBM BG/P)

MESONH is in beta-test (ask Juan ...) you could compile in interactive mode

at CINES on JADE(SGI/ICE)

install the PACKAGE in your `$WORKDIR`

you could compile in interactive mode

backup your work on `/data/$USER`

at ECMWF on c1a (IBM/SP6)

to install MESONH you need more disk space than allowed for 'standard' user (150 mb only ...). Ask to Dominique Lucas, look at this email :

`http://mesonh.aero.obs-mip.fr/cgi-bin/mesonh_interne/`
`mail2html.cgi?file=2009_09_24_15:31:54`

for the compilation use : `llsubmit job_make_mesonh_c1a`

at METEO-FRANCE on yuki/tori

install the PACKAGE in your `$HOME`

for the compilation use : `job_make_mesonh_MFSX8`

2.4 Compiling your own sources only

For your own purposes, you can add new sources or modify the Master sources by building your own library (the User's library). Eventually, some of these modifications may be incorporated in the next release of the Master library, after a merging with modifications from other users.

Suppose you want to create a `MY_MODIF` version.

2.4.1 On PC Linux

1. Prepare your source directory
   Put your own source in a subdirectory of `${SRC_MESONH}/src` named `MY_MODIF`.
   All subdirectories in `MY_MODIF` will be scanned. So if you want you could make a subdirectory for each component of the MESONH Package :

   ```
   cp .../mesonh.f90 MY_MODIF/MNH/.
   cp .../isba.f90   MY_MODIF/SURFEX/.
   ```

   WARNING : In this subdirectory put only fortran source you want to compile. Don't use it as a trash, with old sources file like my_source.f90.old or 'tar' files 'mysource.tar. They will confuse the `make` commande.

2. configure with VER_USER=
   Logout of the current session, to be sure to unset all the environnement variables load with the file `profile_mesonh` of your "master".
   Set the variable `VER_USER` with the name of your "USER VERSION".
   Set also optionnal the `ARCH, VER_MPI`... , you want to use
   Run the "./configure" commande :

   ```
   #export ARCH=...
   #export VER_MPI=...
   export VER_USER=MY_MODIF
   ./configure
   ```

this will regenerate the profile-mesonh file and a copy of this with the extent `profile_mesonh...${VER_USER}` as before
Load it

```
. ../conf/profile_mesnh...${VER_USER}...
```

3. compile your own sources with the command :

```
make user
```

4. install the new binaries in `${SRC_MESONH}/exe` with :

```
make installuser
```

The `make installmaster` need to be done only one time by "version".

Note : As you run the examples, your version should appear in the name of the executables used.

2.4.2 On METEO-FRANCE plateforms

On yuki/tori, you have to make the points 1 and 2. Then in the file `job_make_mesonh_user_MFSX8`, you have to fill the line "export VER_USER=" and then submit the job.

2.5 How to run MESONH on PC linux ?

You have examples of MESONH simulation in the directory `${SRC_MESONH}/MY_RUN/KTEST`.
To run this examples, you have to configure the MESONH package by loading the configuration file `profile_mesonh` you want in directory `${SRC_MESONH}/conf` (necessary if you open a new windows or a new logon).

Chapter 3

The MESONH files :

3.1 The F90 namelists

All the information required to perform a given step of a numerical experiment, are provided by different files including a NAMELIST set. Thus, the Meso-NH user can change the value of the parameters without any compilation (and therefore save computer time). These files provide the way for the Meso-NH user to interact with the numerical code and finally, they contain the identification cards of the different steps of the numerical experiment.

These NAMELISTs are Fortran 90 NAMELISTs, which obey to strict writing rules (Metcalf and Reid 1993): no comment is allowed inside the namelists, no empty namelist can be written (it gives a Fortran execution error).

The information are written in the following form :

```
&NAM_LUNITn  CINIFILE = "     FMFILE.1               " /
&NAM_CONFn  LUSERV = T, LUSERC = F, LUSERR = F, LUSERI = F, LUSERS = F,
LUSERG = F, LUSERH = F, NSV = 0 /
```

`&NAM_LUNITn` is the name of the first namelist of this file, the / character indicates the end of the list of information. These namelists parameters are defined by VAR = VALUE and these prescriptions are separated one from each others by a comma and a blank character. Note that you can use more than one line to give a namelist, but in this case it is imperative to let a blank character at the end of each line.

The Meso-NH user does not need to prescribe all the parameters of a namelist, the missing information are taken equal to the default values written in the fortran code. For example, the second namelist in the previous example can be written as:

```
&NAM_CONFn  LUSERV = T /
```

because the other variables of `&NAM_CONFn` are set to the default values.

In order to clearly separate what can be modified for a given step of a numerical experiment, we affect a different namelist file name for each step.

- To PREpare a Meso-NH file containing PhysioGraphical Datas

  $\Longrightarrow$ file PRE_PGD1.nam

- To PREpare a Meso-NH file with PhysioGraphical Datas in conformity

  $\Longrightarrow$ file PRE_NEST_PGD1.nam

- To PREpare a ZOOMed Meso-NH file with PhysioGraphical Datas

  $\Longrightarrow$ file PRE_ZOOM1.nam

- To PREpare an initial Meso-NH file for an IDEAlized case study

  $\Longrightarrow$ file PRE_IDEA1.nam

- To PREpare an initial Meso-NH file for an REAL case study

  $\Longrightarrow$ file PRE_REAL1.nam

- To SPAWN a Meso-NH file into another one with better horizontal resolution

  $\Longrightarrow$ file SPAWN1.nam

- To EXecute a simulation SEGment for the n^{th} model

  $\Longrightarrow$ file EXSEGn.nam

- To compute DIAGnostics after a simulation

  $\Longrightarrow$ file DIAG1.nam

Because the grid-nesting technique requires the simultanous presence of multiple models in the computer memory, free-parameters must be fixed for every model. This is performed by associating a namelist file per model, this explains why the namelist are suffixed by a number 1 or n just above.

The different parameters present in these files are all given in this book and more details on the description of a given parameter are present in the code itself.

3.2 The Meso-NH files

A Meso-NH FM-file is a set of 2 files, (which can be sticked together via the cpio UNIX command):

- a descriptive part (`.des`) containing information about the file generation and its description (in ASCII characters)

- a binary part (`.lfi`) where the fields are stored. The structure of this file is a direct access type file, written and read by routines developped in Météo-France (Fischer, 1994) based on LFI routines (Clochard, 1989), which can be used on a lot of different computers.

This type of file is used to store all the data necessary to run any step of a numerical experiment. Three different files are taken into account in the Meso-NH project:

- the synchronous file contains all the values of all the fields allowing a restart of the model and of some diagnostic fields desired by the Meso-NH user. All these informations are obtained at the same instant during the simulation,

  thus they are synchronous.

- the diachronic file contains time series of information desired by the Meso-NH user. They are obtained during more than one time step of the model. It is the format in which your file must be if you want to plot it with the graphics software `diaprog` (you can convert a synchronous file into a diachronic one with `conv2dia`).

- the physiographic file contains external information like orography, vegetation classes, chemical emissions, data sets, etc.

3.2.1 The synchronous file

This type of file contains only information corresponding to the same instant of the simulation, it remains open during a whole time step of the simulation, and the writing orders can be given from any routine of the model.

The descriptive part

This part is the list of all the namelists of the EXSEG$n.nam file. Thus, a complete description of this part is given with the EXSEG$n.nam description in chapter 9.

If the file has been generated during a segment of the model integration, the `.des` part contains the different namelists fixing the free-parameters for the dynamics and the physics of the Meso-NH model. This allows the user to know a large part of the history of this file. For the namelists or variables ommited in the EXSEG$n.nam file, the values are set to the default ones (see the tables in ch.9).

If the file is the result of the initialization programs (`PREP_IDEAL_CASE`, `PREP_REAL_CASE` or `SPAWNING`), the values of the namelists variables are the ones of the descriptive part of the input

file of the program if it does exist. Otherwise, the values are set to the default ones, except for these that can be initialized during the initialization program (e.g. `CINIFILE` or `LUSERV`).

Note that a physiographic file does not have a descriptive part.

The binary part

All the writings and readings of this type of files are done through LFI routines. A general subroutine to read and write a Meso-NH file is given in the Meso-NH library, it provides a file including the fields of the previous record list. This Fortran library provides a way to tackle direct access binary files and thus a very quick access to the data stored in this file in any order.

It should be noted that supplementary fields can be added to these basic information, which have been obtained at the same instant. In order to be easily drawn by the Meso-NH graphic package, the commentar field must be filled, according to the following rules:

- the length of the character string is equal to 100
- the type of the supplementary field must be specified :

type	commentar field
3D scalar	X_Y_Z_varname (UNIT)
2D scalar	X_Y_varname (UNIT)
3D vector	VX_xvarname_VY_yvarname_VZ_zvarname (UNIT)
2D scalar	VX_xvarname_VY_yvarname_VZ_zvarname (UNIT) or VX_xvarname_VY_yvarname (UNIT)
1D scalar	Z_zvarname (UNIT)

3.2.2 The diachronic file

It is a file obtained during a segment of simulation or resulting of the conversion of a synchronous file with `conv2dia` for graphical purposes.

The file directly obtained during the simulation has a name ended by `.000`, and contains records such as averaged variables, tendencies, fluxes stored at different times of the simulation on the whole or some parts of the domain. Such records are obtained by asking for temporal series (9.5.1), budgets (9.2.3), aircraft or balloon (9.4), profiler or station (9.5.2), LES diagnostics (C).

3.2.3 The physiographic file

It is a bidimensional MesoNH file with contains surface data as orography, vegetation classes, chemical emissions, etc.

See the documentation :“THE EXTERNALIZED SURFACE USER’S GUIDE” for more details.

3.3 References

J. Clochard, 1989: Logiciel de Fichiers Indexés. Direction de la Mtorologie Nationale. Note de travail ARPEGE n°12.

J. Clochard, 1991: Logiciel de Fichiers Indexés. Direction de la Mtorologie Nationale. Technical report.

D. Gazen, 1999: Parallel IO routines. Man page on Meso-NH web site.

C. Fischer, 1994: File structure and content in the Meso-NH model. Meso-NH note.

Chapter 4

Creation of MESO-NH physiographic data file

4.1 PREP_PGD

The physiographic fields are averaged or interpolated on the MESO-NH physiographic grid by the program **PREP_PGD**. They are stored in a FM file, called PGD file, but with fewer elements than a MESO-NH file. With the physiographic 2D fields, the geographic and grid data are written in this file. Note that a MESO-NH simulation runs on the grid defined here in **PREP_PGD**.

4.1.1 Namelist NAM_PGDFILE

Fortran name	Fortran type	default value
CPGDFILE	character (LEN=28)	' '
NHALO	INTEGER	15

- CPGDFILE : name of the output Physiographic Data File
- NHALO : Size of the halo for parallel distribution

4.1.2 Namelists for the externalized surface

As indicated above, further definition of surface parameters are not done by MESONH itself, but by the externalized surface included in it. For more informations see SURFEX documentation.

- NAM_PGD_SCHEMES
- NAM_PGD_GRID
- NAM_CONF_PROJ

- NAM_CONF_PROJ_GRID
- NAM_INIFILE_CONF_PROJ
- NAM_WRITE_COVER_TEX
- NAM_READ_DATA_COVER
- NAM_PGD_ARRANGED_COVER
- NAM_COVER
- NAM_ECOCLIMAP2
- NAM_ZS
- NAM_SEABATHY
- NAM_ISBA
- NAM_DATA_FLAKE
- NAM_DUMMY_PGD
- NAM_CH_EMIS_PGD

4.1.3 Examples of PRE_PGD1.nam file

A **PREP_PGD** run where you use the data files provided by the MESO-NH team and some files of your own for the dummy fields:

```
&NAM_PGDFILE        CPGDFILE = 'PGDFILE_1' /
&NAM_PGD_GRID       CGRID = 'CONF PROJ ' /
&NAM_CONF_PROJ      XLAT0 = 45., XLON0=0., XRPK=0.7, XBETA=0. /
&NAM_CONF_PROJ_GRID NIMAX=100, NJMAX=100, XLATCEN=42.5, XLONCEN=2.5, XDX=10000. , XDY=10000. /
&NAM_INIFILE_CONF_PROJ /
&NAM_PGD_SCHEMES CNATURE='ISBA  ', CSEA='SEAFLX', CTOWN='TEB   ', CWATER='WATFLX' /
&NAM_COVER YCOVER='ecoclimap_v2', YFILETYPE='DIRECT' /
&NAM_ZS    YZS='gtopo30', YFILETYPE='DIRECT' /
&NAM_ISBA  YCLAY='clay_fao', YCLAYFILETYPE='DIRECT',  XUNIF_RUNOFFB=0.5 ,
           YSAND='sand_fao', YSANDFILETYPE='DIRECT' /
&NAM_DUMMY_PGD  NDUMMY_NBR         = 6
                CDUMMY_NAME(1)     = 'SST_2001062100'
                CDUMMY_AREA(1)     = 'SEA'
```

```
CDUMMY_ATYPE(1)    = 'ARI'
CDUMMY_FILE(1)     = 'SSTn2001062100.dat'
CDUMMY_FILETYPE(1) = 'ASCLLV'
CDUMMY_NAME(2)     = 'SST_2001062200'
CDUMMY_AREA(2)     = 'SEA'
CDUMMY_ATYPE(2)    = 'ARI'
CDUMMY_FILE(2)     = 'SSTn2001062200.dat'
CDUMMY_FILETYPE(2) = 'ASCLLV'
CDUMMY_NAME(3)     = 'SST_2001062300'
CDUMMY_AREA(3)     = 'SEA'
CDUMMY_ATYPE(3)    = 'ARI'
CDUMMY_FILE(3)     = 'SSTn2001062300.dat'
CDUMMY_FILETYPE(3) = 'ASCLLV'
CDUMMY_NAME(4)     = 'SST_2001062400'
CDUMMY_AREA(4)     = 'SEA'
CDUMMY_ATYPE(4)    = 'ARI'
CDUMMY_FILE(4)     = 'SSTn2001062400.dat'
CDUMMY_FILETYPE(4) = 'ASCLLV'
CDUMMY_NAME(5)     = 'SST_2001062500'
CDUMMY_AREA(5)     = 'SEA'
CDUMMY_ATYPE(5)    = 'ARI'
CDUMMY_FILE(5)     = 'SSTn2001062500.dat'
CDUMMY_FILETYPE(5) = 'ASCLLV'
CDUMMY_NAME(6)     = 'SST_2001062600'
CDUMMY_AREA(6)     = 'SEA'
CDUMMY_ATYPE(6)    = 'ARI'
CDUMMY_FILE(6)     = 'SSTn2001062600.dat'
CDUMMY_FILETYPE(6) = 'ASCLLV'/
```

Another **PREP_PGD** run for the father PGD file, with all the namelist variables :

```
&NAM_PGDFILE CPGDFILE='PGD_AMMA1_10km_m46_b1' /
&NAM_PGD_SCHEMES  CNATURE='ISBA' , CSEA='SEAFLX', CWATER='WATFLX', CTOWN='TEB'/
&NAM_PGD_GRID CGRID='CONF PROJ'/
&NAM_CONF_PROJ   XLAT0=13. ,XLON0=-1.5, XRPK=0.2249510543, XBETA=0 /
&NAM_CONF_PROJ_GRID  XLATCEN=13. , XLONCEN=-1.5 , NIMAX=324 ,NJMAX=240 ,
                     XDX=10000 , XDY=10000/
&NAM_COVER      YCOVER='ecoclimats_v2', YFILETYPE='DIRECT',LRM_TOWN=.FALSE. /
&NAM_ZS         YZS='gtopo30',    YFILETYPE='DIRECT' ,
                COROGTYPE='AVG', XENV=0., NZSFILTER=1 /
&NAM_ISBA       NPATCH=12, CISBA='3-L',  CPHOTO='AGS', NGROUND_LAYER=3,
                YCLAY='clay_fao', YCLAYFILETYPE='DIRECT' ,
                YSAND='sand_fao', YSANDFILETYPE='DIRECT',
                XUNIF_RUNOFFB=0.5 /
&NAM_CH_EMIS_PGD NEMIS_PGD_NBR=0 /
&NAM_DUMMY_PGD  NDUMMY_PGD_NBR=0 /
```

And for the son PGD file :

```
&NAM_PGDFILE CPGDFILE='PGD_AMMA1_5km_m46_b1' /
&NAM_PGD_SCHEMES  CNATURE='ISBA' , CSEA='SEAFLX', CWATER='WATFLX', CTOWN='TEB'/
&NAM_PGD_GRID YINIFILE='PGD_AMMA1_10km_m46_b1',YFILETYPE='MESONH' /
&NAM_INIFILE_CONF_PROJ IXOR=101 ,IYOR=21 ,IXSIZE=180 ,IYSIZE=150 ,
                       IDXRATIO=2, IDYRATIO=2/
&NAM_COVER      YCOVER='ecoclimats_v2', YFILETYPE='DIRECT',LRM_TOWN=.FALSE. /
&NAM_ZS         YZS='gtopo30', YFILETYPE='DIRECT' ,
                COROGTYPE='AVG', XENV=0., NZSFILTER=1 /
&NAM_ISBA       NPATCH=12, CISBA='3-L',  CPHOTO='AGS', NGROUND_LAYER=3,
                YCLAY='clay_fao', YCLAYFILETYPE='DIRECT' ,
                YSAND='sand_fao', YSANDFILETYPE='DIRECT',
                XUNIF_RUNOFFB=0.5 /
&NAM_CH_EMIS_PGD NEMIS_PGD_NBR=0 /
&NAM_DUMMY_PGD  NDUMMY_PGD_NBR=0 /
```

4.2 Modification of PGD files for grid-nesting: PREP_NEST_PGD

In order to run models with the gridnesting technique, a condition on the orography must be satisfied. In the following, if file #2 is completely included in (and therefore in interaction during the run with) file #1, file #2 will be called the SON file, and file #1 the DAD file. In the following, the DAD file number must be smaller than any of its SON number.

The condition on the orography is: "the mean of orography for a SON file in the domain corresponding to the grid mesh of its DAD file, must be equal to the orography of the DAD file in this mesh".

Such a condition is not automatically satisfied when using enhanced orographies. The program **PREP_NEST_PGD** performs post-treatments on the orographies of up to 8 PGD files that will be used to create initialization files for a gridnested run. It modifies the orography of a DAD from the mean of the orography of its (several) SON(s).

The namelist file PRE_NEST_PGD1.nam contains:

1. Namelists NAM_PGD**N** (where **N** goes from 1 to 8):

   - YPGD**N**: name of the PGD file #**N**
   - IDAD: number of the DAD file of file #**N**. The DAD file number IDAD must be smaller than #**N**.

2. Namelist NAM_NEST_PGD:

   - YNEST: string of 2 characters to be added to the PGD file names to define the corresponding output PGD file names. The input file *YPGD***N** will be modified into file *YPGD***N**.nest*YNEST*

Example of namelist PRE_NEST_PGD1.nam:

```
&NAM_PGD1 YPGD1 = 'PGDFILE_1' /
&NAM_PGD2 YPGD2 = 'PGDFILE_2', IDAD = 1 /
&NAM_PGD3 YPGD3 = 'PGDFILE_3', IDAD = 1 /
&NAM_PGD4 YPGD4 = 'PGDFILE_4', IDAD = 3 /
&NAM_PGD5 YPGD5 = 'PGDFILE_5', IDAD = 2 /
&NAM_PGD6  /
&NAM_PGD7  /
&NAM_PGD8  /
&NAM_NEST_PGD YNEST = 'e1' /
```

4.3 Zoom of a PGD file: ZOOM_PGD

The previous condition on the orography needed when using the gridnesting technique implies that all the PGD files have to be created (with PREP_PGD and PREP_NEST_PGD programs) **before** beginning the run. However, the user is not always sure where (and when) to initialize the inner models. To avoid to set exactly the domain of inner models at the PREP_PGD step, one solution is to make PGD file on larger domain and then, zoom it[1] on the part of the domain of interest when known with the following program **ZOOM_PGD**. Then the output PGD file is used as PGD file for the interpolations of atmospheric fields with SPAWNING and PREP_REAL_CASE programs.

The namelist file PRE_ZOOM1.nam contains 2 namelists:

1. Namelist NAM_PGDFILE: (contains file names)

Fortran name	Fortran type	default value
CPGDFILE	character (LEN=28)	'PGDFILE'
YZOOMFILE	character (LEN=28)	none
YZOOMNBR	character (LEN=2)	'00'

- CPGDFILE : name of the input Physiographic Data File
- YZOOMFILE : optional name of the zoomned FM-file (output file). If the user does not specify this name, or if YZOOMFILE = CPGDFILE, the code builds the zoomned FM-file name as:

  YZOOMFILE = CPGDFILE.zYZOOMNBR
- YZOOMNBR : : NumBeR which will be added to CPGDFILE to generate the name of the Zoomned FM-file (string of 2 characters).

2. Namelist NAM_MESONH_DOM: (contains domain definition variables)

Fortran name	Fortran type	default value	remarks
NIMAX	integer	input PGD domain	
NJMAX	integer	input PGD domain	
NXOR	integer	NUNDEF	if NUNDEF, domain is in the middle of PGD
NYOR	integer	NUNDEF	if NUNDEF, domain is in the middle of PGD

- NIMAX : number of grid points in I direction, according to input file grid, recovered by the new domain. NIMAX must be equal to $2^m \times 3^n \times 5^p$ with $(m, n, p) \in [0, +\infty[$

[1]This was done during the PREP_IDEAL_CASE or PREP_REAL_CASE step before the masdev4_6 version.

- NJMAX : number of grid points in J direction, according to input file grid, recovered by the new domain. NJMAX must be equal to $2^m \times 3^n \times 5^p$ with $(m, n, p) \in [0, +\infty[$
- NXOR : first point I index, left to and out of the new physical domain.
- NYOR : first point J index, under and out of the new physical domain.

Example of namelist PRE_ZOOM1.nam:

```
&NAM_PGDFILE CPGDFILE = 'PGDFILE_1.neste1' ,
             YZOOMNBR = '58'    /
&NAM_MESONH_DOM NIMAX=60, NJMAX=50,
                NXOR=5, NYOR=8      /
```

Chapter 5

Preparation of an ideal simulation : PREP_IDEAL_CASE

5.1 Overview of PREP_IDEAL_CASE functionalities

The "PREP_IDEAL_CASE" program prepares a MESONH file, that contains all the parameters and fields necessary for the execution of the MESONH model. Specifically, the grid parameters, the initial fields and the geophysical fields are included in this file. It is possible using this program to generate idealized fields defined by few parameters.

The generated initial conditions are produced analytically, leading to quasi-1D fields or 3D fields or a single profile build with either:

- layers of constant Brunt-Vaisala frequency, shear and humidity
- a Radiosounding and ideal surface fields
- a Radiosounding and real physiographic fields
- a Radiosounding and real and ideal surface fields at the same time

For these latter cases, the initial fields may be hydrostatically or geostrophically balanced or not. For these fields to satisfy the anelastic constraint, a final correction is applied to them.

The interaction between the PREP_IDEAL_CASE program and the user is made through the PRE_IDEA1.nam file. The degrees of freedom are collected in a set of namelists, read by this program.

5.2 The input: the PRE_IDEA1.nam file

It is made of two parts :

- A namelist-part with directives for the preparation of an idealized case (always present). The order of namelists is free and unset namelists can be ommited.
- A free-formatted part describing a vertical profile of n layers of constant moist Brunt-Vaisala frequency or a radiosounding and sometimes the explicit list of the heights of the vertical levels. This part can be present or absent in the other cases.

To initialize a simulation with a radiosounding and real terrain conditions, it is necessary to perform the PREP_PGD program to create a MESO-NH physiographic data file. This data file contains the orography and the physiographic data fields (related to the soil scheme). It is also possible to perform a complete ideal case with ideal orography and non trivial surface conditions. The user can combine the two possibilities with flags included in the namelist NAM_REAL_PGD and initialize a simulation with a real orography and idealized homogeneous surface fields. If a PREP_PGD file is specified and if the flags in namelist NAM_REAL_PGD are set to FALSE, homogeneous values can be imposed by the user in namelists from the externalized surface facility PGD (namelists NAM_COVER and NAM_ISBA), else the PREP_PGD fields are taken into account.

In the following, the namelists are listed in alphabetical order.

5.2.1 Namelist NAM_AERO_PRE (init. aerosol scalar variables)

If you initialize aerosol during PREP_IDEAL_CASE as for ORILAM (chemical aerosols), DUST and SEA SALT. use the following namelist variables:

Fortran name	Fortran type	default value
LORILAM	logical	FALSE
LDUST	logical	FALSE
LSALT	logical	FALSE
LINITPM	logical	FALSE
XINIRADIUSI	real	0.05
XINIRADIUSJ	real	0.2
XINISIGI	real	1.8
XINISIGJ	real	2.0
XN0IMIN	real	10.
XN0JMIN	real	1.
CRGUNIT	character	"NUMB"
NMODE_DST	integer	3
XN0MIN	real	1.e3 , 1.e1 , 1.e-2
XINIRADIUS	real	0.044, 0.3215, 1.575
XINISIG	real	2.0, 1.78, 1.85
CRGUNITD	character	"NUMB"
NMODE_SLT	integer	3
XN0MIN_SLT	real	1.e4 , 1.e2 , 1.e-1
XINIRADIUS_SLT	real	0.14, 1.125, 7.64
XINISIG_SLT	real	1.9, 2., 2.
CRGUNITS	character	"MASS"

- LORILAM: Flag to activate chemical aerosol initialization (only if LCH_INIT_FIELD=T in NAM_CH_MNHCn_PRE).
- LDUST: Flag to activate passive dust initialization (3 modes).
- LSALT: Flag to activate passive sea salt initialization (3 modes).
- LINITPM: Flag to activate primary aerosol initialization (Black and Organic carbon) from concentration of CO (only if LORILAM=T in NAM_CH_MNHCn_PRE).
- XINIRADIUSI: Initial mean radius of aitken mode in μm (only if LORILAM=T in NAM_AERO_PRE).
- XINIRADIUSJ: Initial mean radius of accumulation mode in μm (only if LORILAM=T in NAM_AERO_PRE).
- XINISIGI: Initial standard deviation of aitken mode (only if LORILAM=T in NAM_AERO_PRE).
- XINISIGJ: Initial standard deviation of accumulation mode (only if LORILAM=T in NAM_AERO_PRE).

- XN0IMIN: Minimum number concentration of aitken mode (only if LORILAM=T in NAM_AERO_PRE).
- XN0JMIN: Minimum number concentration of accumulation mode (only if LORILAM=T in NAM_AERO_PRE).
- CRGUNIT: Definition of XINIRADIUSI or XINIRADIUSJ: mean radius is in mass or in number; possible values are MASS or NUMB (only if LORILAM=T in NAM_AERO_PRE).
- NMODE_DST: Number of DUST mode (between 1 and 3 and only if LDUST=T in NAM_AERO_PRE).
- XN0MIN: Minimum number concentration of the NMODE_DST in particles by m3 (only if LDUST=T in NAM_AERO_PRE).
- XINIRADIUS: Initial mean radius of the NMODE_DST modes in μm (only if LDUST=T in NAM_AERO_PRE).
- XINISIG: Initial standard deviation of the NMODE_DST modes (only if LDUST=T in NAM_AERO_PRE).
- CRGUNITD: Definition of XINIRADIUS : mean radius is in mass or in number; possible values are MASS or NUMB (only if LDUST=T in NAM_AERO_PRE).
- NMODE_SLT: Number of SALT mode in μm (between 1 and 3 and only if LSALT=T in NAM_AERO_PRE).
- XN0MIN_SLT: Minimum number concentration of the NMODE_SLT in particles by m3 (only if LSALT=T in NAM_AERO_PRE).
- XINIRADIUS_SLT: Initial mean radius of the NMODE_SLT modes (only if LSALT=T in NAM_AERO_PRE).
- XINISIG_SLT: Initial standard deviation of the NMODE_SLT modes (only if LSALT=T in NAM_AERO_PRE).
- CRGUNITS: Definition of XINIRADIUS_SLT : mean radius is in mass or in number; possible values are MASS or NUMB (only if LSALT=T in NAM_AERO_PRE).

5.2.2 Namelist NAM_BLANK (available variables)

Fortran name	Fortran type	default value
XDUMMY1 .. XDUMMY8	real	0.
NDUMMY1 .. NDUMMY8	integer	0
LDUMMY1 .. LDUMMY8	logical	TRUE
CDUMMY1 .. CDUMMY8	80 characters	”
XDUMMY	array(real)	20* 0.
NDUMMY	array(integer)	20* 0
LDUMMY	array(logical)	20* TRUE
CDUMMY	array(80 characters)	20* ”

Eight dummy, real, integer, logical and character*80 variables and arrays of dummy, real, integer, logical and character*80 for test and debugging purposes are defined and passed through the namelist read operations. None of the MesoNH routines uses any of these variables. When a developper choses to introduce temporarily a parameter to some subroutine, he has to introduce a USE MODD_BLANK statement into that subroutine. Then he can use any of the variables defined here and change them easily via the namelist input.

5.2.3 Namelist NAM_CH_MNHCn_PRE (init. chemistry scalar variables)

If you initialize MNH-C using PREP_IDEAL_CASE, use the following namelist variables:

Fortran name	Fortran type	default value
LCH_INIT_FIELD	logical	FALSE
CCHEM_INPUT_FILE	80 characters	MNHC.input

- LCH_INIT_FIELD: Flag to activate initialization subroutine CH_INIT_FIELD.
- CCHEM_INPUT_FILE: name of the general purpose input file for initialization.

5.2.4 Namelist NAM_CONF_PRE (configuration variables)

Fortran name	Fortran type	default value
LCARTESIAN	logical	TRUE
LPACK	logical	TRUE
CEQNSYS	3 characters	’DUR’
NVERB	integer	5
CIDEAL	4 characters	’CSTN’
CZS	4 characters	FLAT
LBOUSS	logical	FALSE
LPERTURB	logical	FALSE
LFORCING	logical	FALSE
LSHIFT	logical	FALSE

- LCARTESIAN : Flag for cartesian geometry
  - .TRUE. for cartesian geometry

  - .FALSE. for conformal projection

- LPACK : Flag to compress FM file for 1D or 2D version.
- CEQNSYS : Equation system resolved by the MESONH model
  - 'LHE' Lipps and HEmler anelastic system
  - 'DUR' approximated form of the DURran version of the anelastic sytem
  - 'MAE' classical Modified Anelastic Equations but with not any approximation in the momentum equation
- NVERB : verbosity level
  - 0 for minimum of prints
  - 5 for intermediate level of prints
  - 10 for maximum of prints.

  If $CSURF$="EXTE" in namelist NAM_GRn_PRE , NVERB=10 prints two LaTeX files containing the initialisation of surface scheme variables for each type of surface cover (in french or in english).
- CIDEAL : kind of idealized fields
  - 'CSTN' : Constant moist Brunt Vaisala frequency case
  - 'RSOU' : radiosounding case
- CZS : orography selector The formulae are given below in the description of the namelist NAM_GRIDH_PRE.
  - 'FLAT' : constant XHMAX orography (zero by default)
  - 'SINE' : sine-shaped orography
  - 'BELL' : bell-shaped orography
  - 'DATA': discretized orography. The data describing the orography are given in the free format part. Only the orography corresponding to the computational domain must be provided in free format. For 3D orography, data are read like if it was a map (the first line is the Northern border and the first data is the North-West corner) with one line per Y-axis increment.
- LBOUSS : Flag for a Boussinesq version.

– .TRUE. The reference anelastic state is $\theta_{ref} = cte = \theta_{ref}(z = 0)$ and $\rho_{ref} = cte = \rho_{ref}(z = 0)$. In this case, the stratification is taken into account in the Meso-NH model in the flottability term. The typical length, on which this stratification varies, is much greater than the domain heigth and the θ_{ref} variation can be therefore neglected.

– .FALSE. The reference anelastic state varies with the altitude.

- LPERTURB : Flag to add a perturbation on the initially horizontally homogeneous fields. This perturbation is not balanced.

  3 perturbation types are implemented in the routine *set_perturb.f90* :

  – a spherical perturbation on the dry potential temperature and the moisture fields (typical for convection initialization).

  – a perturbation on the horizontal components of the wind derived from a streamfunction (typical for large scale studies). This prevents the wind from becoming divergent.

  – a perturbation on the dry potential temperature field at the first mass level near the ground, corresponding to a white noise (uniform amplitude in the spectral space) (typical for Large Eddy Simulations initialization)

  When set to .TRUE., the parameters for the exact definition of the perturbation can be set in the namelist NAM_PERT_PRE or sometimes can be modified directly in the subroutine *set_perturb.f90*

- LFORCING : Flag to specify forcing sources. When .TRUE., the precise definition of the forcing is set in the free-format part of PRE_IDEA1.nam (see 5.4.4).

- LSHIFT : flag to shift altitudes in boundary layer. If LGEOSBAL=TRUE, LSHIFT will be set to FALSE.

5.2.5 Namelist NAM_CONFn (configuration variables for modeln)

Fortran name	Fortran type	default value
LUSERV	logical	TRUE
LUSERC	logical	FALSE
LUSERI	logical	FALSE
NSV_USER	integer	0

(see 5.5.2 for more details for these cases)

- LUSERV : Flag to write r_v (vapor mixing ratio) in initial file. It is reset to .TRUE. when CIDEAL ='RSOU' or 'CSTN'. This has been done in order to avoid to treat the dry case as a particular case but as a moist case with humidity equal to 0.

- LUSERC : Flag to write r_c (cloud mixing ratio) in initial file. This case is only allowed when CIDEAL ='RSOU' (radiosounding case) and KIND='PUVTHDMR' or KIND='ZUVTHLMR'
- LUSERI : Flag to write r_i (ice mixing ratio) in initial file. This case is only allowed when CIDEAL ='RSOU' (radiosounding case) and KIND='PUVTHDMR'
- NSV_USER : number of scalar variables Note that if NSV_USER is different from 0, the Scalar Variables are initialized to 0 by the program

5.2.6 Namelist NAM_CONFZ (configuration variables for splitting along z)

Fortran name	Fortran type	default value
NZ_VERB	integer	0
NZ_PROC	integer	0
NZ_PROCIO_R	integer	1
NZ_PROCIO_W	integer	1
MPI_BUFFER_SIZE	integer	40
LMNH_MPI_BSEND	logical	TRUE
LMNH_MPI_ALLTOALLV_REMAP	logical	FALSE
NZ_SPLITTING	integer	10

- NZ_VERB: level of message for NZ solver and I/O
- NZ_PROC: number of processors to use in the Z splitting. The default value (0) yields an automatic calculation of the number.
- NZ_PROCIO_R: number of processors to use for parallel I/O when reading file. The default value (1) yields a reading from 1 file only. If more than 1 file, the 3D field are written as several 2D slides.
- NZ_PROCIO_W: Number of processors to use for parallel I/O when writing file. The default value (1) yields a writing into 1 file only. If more than 1 file, the 3D field are written as several 2D slides.
- MPI_BUFFER_SIZE: default size for MPI_BSEND buffer in 10^6 bytes. MPI_BUFFER_SIZE corresponds approximately to the size of the domain, that is, NX*NY*NZ for I/O in 1 file, and NX*NY for I/O in N 2D-slide files.
- LMNH_MPI_BSEND: during HALO exchange and FFT transposition, switch to use bufferized either MPI_BSEND routine or asynchrone MPI_ISEND routine. Depending on the computer and size of the problem, one or the other option could run faster. MPI_ISEND also uses less memory so MPI BUFFER SIZE should be decreased.
- LMNH_MPI_ALLTOALLV_REMAP:

- FALSE: FFT remap with send/recv <=> NZ_SPLITTING=10
- TRUE: FFT remap with mpi_alltoallv <=> NZ_SPLITTING=14 (BG/MPICH optimization)

- NZ_SPLITTING: setting by namelist for debugging by expert user only. The non-expert user will use LMNH_MPI_ALLTOALLV_REMAP=T/F only: IZ=1=flat_inv; IZ=2=flat_invz; IZ=1+2=the two; +8=P1/P2.

5.2.7 Namelist NAM_DIMn_PRE (contains dimensions)

Fortran name	Fortran type	default value
NIMAX	integer	10
NJMAX	integer	10

- NIMAX : number of mass points in x-direction of the initial file is $NIMAX + 2JPHEXT$ ($JPHEXT$ corresponds to the number of marginal points in the horizontal directions and is fixed to 1 for the present Meso-NH version). NIMAX must be equal to $2^m \times 3^n \times 5^p$ with $(m, n, p) \geq 0$

- NJMAX : number of mass points in y-direction of the physical domain. The total size of the array written in the initial file is $NJMAX + 2JPHEXT$. NJMAX must be equal to $2^m \times 3^n \times 5^p$ with $(m, n, p) \geq 0$

5.2.8 Namelist NAM_DYNn_PRE (pressure solver)

Fortran name	Fortran type	default value
CPRESOPT	5 characters	'CRESI'
NITR	integer	4
XRELAX	real	1.
LRES	logical	.FALSE.
XRES	real	1.E-07

- CPRESOPT : gives the type of pressure solver used for the elliptic equation ('RICHA', 'CGRAD', 'CRESI'). This equation is solved in order to ensure the anelastic constraint for the initial wind field. Note that the solver is applied even for the FLAT case when the Earth spericity is taken into account.

- NITR : number of iterations used for the resolution of the elliptic equation (solver = "CPRESOPT").

- XRELAX : relaxation factor used by the Richardson method (CPRESOPT = "RICHA").

- LRES : flag to change the residual divergence limit

- XRES : Value of the residual divergence limit

5.2.9 Namelist NAM_GRID_PRE (grid definition)

Fortran name	Fortran type	default value
XLON0	real	0.
XLAT0	real	60.
XBETA	real	0.
XRPK	real	1.
XLONORI	real	350.
XLATORI	real	37.

- XLON0 : reference longitude for conformal projection and cartesian plane (if LCARTESIAN =.TRUE. this value can be usefull to compute local solar time)
- XLAT0 : reference latitude for conformal projection and cartesian plane
- XBETA : rotation angle for conformal projection and cartesian plane
- XRPK : cone factor for the projection (only if LCARTESIAN =.FALSE.):
  - XRPK=1: polar stereographic projection from south pole
  - 1>XRPK>0: Lambert projection from south pole
  - XRPK=0: Mercator projection from earth center
  - -1<XRPK<0: Lambert projection from north pole
  - XRPK=-1: polar stereographic projection from north pole
- XLONORI : Longitude (in degrees) of the origine point (not used if LCARTESIAN =.TRUE.). This point is the mass point of conformal coordinates (x=0,y=0) of the Meso-NH grids (See annexe D for more details on the Meso-NH grids).
- XLATORI : Latitude (in degrees) of the origine point (not used if LCARTESIAN =.TRUE.)

5.2.10 Namelist NAM_GRIDH_PRE (horizontal grid definition)

Fortran name	Fortran type	default value
XLATCEN	real	XUNDEF
XLONCEN	real	XUNDEF
XDELTAX	real	5000.
XDELTAY	real	5000.
XHMAX	real	300. / 0.
NEXPX	integer	3
NEXPY	integer	1
XAX	real	10000.
XAY	real	10000.
NIZS	integer	5
NJZS	integer	5

- XLATCEN : latitude of the center of the domain for initialization. This point is vertical vorticity point (See annexe D for more details on the Meso-NH grids)

- XLONCEN : longitude of the center of the domain for initialization. This point is vertical vorticity point (See annexe D for more details on the Meso-NH grids)

- XDELTAX : mesh length (in meters) in x-direction on the conformal or cartesian plane. It is not used if you read informations in a Meso-NH constant file (PGD_FILE).

- XDELTAY : mesh length (in meters) in y-direction on the conformal or cartesian plane. It is not used if you read informations in a Meso-NH constant file (PGD_FILE).

- XHMAX[1] : Maximum height (in meters) h_{max} for orography (case CZS $\neq$ 'FLAT') or ground level for flat orography

- NEXPX : Exponent exp_x for orography in case of CZS='SINE'

- NEXPY : Exponent exp_y for orography in case of CZS='SINE'

- XAX : Widths (in meters) a_x along x for orography in case CZS='BELL'

$$z_s\left(\hat{x},\hat{y}\right) = \frac{h_{max}}{\left(1+\left(\frac{\hat{x}-NIZS*XDELTAX}{XAX}\right)^2+\left(\frac{\hat{y}-NJZS*XDELTAX}{XAY}\right)^2\right)^{1.5}}$$

  in the three-dimensional case.

$$z_s\left(\hat{x}\right) = \frac{h_{max}}{1+\left(\frac{\hat{x}-NIZS*XDELTAX}{XAX}\right)^2}$$

  in the two-dimensional case.

- XAY : Width (in meters) a_y along y for orography in case CZS='BELL'

- NIZS : Localization in x-direction of the mountain center in the case CZS ='BELL'. ($x_s = NIZS * XDELTAX$) It refers to a vertical velocity point at the ground (NIZS, NJZS)(See annexe D for more details on the Meso-NH grids)

- NJZS : Localization in y-direction of the mountain center in the case CZS ='BELL'. ($y_s = NJZS * XDELTAY$)

[1] default is 300. for mountain and 0 for flat orography

5.2.11 Namelist NAM_GRn_PRE (surface scheme choice)

Fortran name	Fortran type	default value
CSURF	4 characters	"NONE"

- CSURF : ground selector.

  – 'NONE' no surface scheme will be activated during the future MesoNH simulation, we therefore do not need any surface parameters. All the namelists of the externalized surface will be ignored.

  – 'EXTE' the externalized surface is used. See the SURFEX documentation for more details.

5.2.12 Namelist NAM_LBCn_PRE (lateral boundary conditions)

Fortran name	Fortran type	default value
CLBCX	array(2 characters)	2*"CYCL"
CLBCY	array(2 characters)	2*"CYCL"

- CLBCX : represent the type of lateral boundary condition at the left and right boundaries along x (CLBCX(1) and CLBCX(2) respectively). Possible values are "CYCL", "OPEN", "WALL" for cyclic, open and rigid wall boundary conditions respectively. It should be note that CLBCX(1) or CLBCY(1) refers to the lowest index values (IIB , IJB for X and Y directions) and CLBCX(2) or CLBCY(2) to the highest index values (IIE and IJE). Please note that :

  $$CLBCo(1) = "CYCL" \Rightarrow CLBCo(2) = "CYCL"$$

  The same boundary conditions must be used for the MESO-NH run itself (see EXSEG1.nam namelist) Note also that CYCLIC conditions are not possible with a PGD file (CPGD_FILE different to ' ' in NAM_REAL_PGD).

- CLBCY : same as CLBCX but for the left and right boundaries along y (CLBCY(1) and CLBCY(2) respectively). They are strings of 4 characters.

5.2.13 Namelist NAM_LUNITn (logical unit names)

Fortran name	Fortran type	default value
CINIFILE	28 characters	'INIFILE'

- CINIFILE : name of the initial FM-file produced by PREP_IDEAL_CASE, it will then be used as initial file in a MESONH numerical simulation.

5.2.14 Namelist NAM_PERT_PRE (set analytical perturbations)

Fortran name	Fortran type	default value
CPERT_KIND	characters	'TH'
XAMPLITH	real	1.5
XAMPLIRV	real	0.0
XAMPLIUV	real	1.0834
XAMPLIWH	real	0.1
NKWH	integer	2
LSET_RHU	logical	TRUE
XCENTERZ	real	2000.
XRADX	real	10000.
XRADY	real	10000.
XRADZ	real	2000.

- CPERT_KIND: Defines the type of the perturbation
  - 'TH' : thermodynamical fields perturbation (θ and r_v)
  - 'UV' : horizontal wind fields perturbation (U and V)
  - 'WH' : white noise applied to θ
  - 'WW' : white noise applied to wind components
- XAMPLITH: maximum perturbation for θ
- XAMPLIRV: maximum perturbation for r_v
- XAMPLIUV: maximum perturbation for U and V
- XAMPLIWH: maximum perturbation for the normalized white noise (temperature or wind)
- NKWH: Upper level of the layer starting from the ground where the white noise is applied
- LSET_RHU: Conservation of the relative humidity
  - TRUE the relative humidity is conserved in the θ perturbation
  - FALSE the r_v perturbation is computed with the XAMPLIRV amplitude
- XCENTERZ: Height of the maximum of the θ perturbation (m)
- XRADX: radius of the perturbation along X (m)
- XRADY: radius of the perturbation along Y (m)
- XRADZ: radius of the perturbation along Z (m)

5.2.15 Namelist NAM_REAL_PGD (PGD file flags)

Fortran name	Fortran type	default value
CPGD_FILE	characters	' '
LREAD_ZS	logical	FALSE
LREAD_GROUND_PARAM	logical	FALSE

- CPGD_FILE : name of the physiographic data file containing the ground data fields. The file must be generated by the PRE_PGD program. **For a purely ideal case, the CPGD_FILE variable may be deleted from the namelist or set to its default value ' '. The horizontal grid will be read in the PGD file and therefore, the mesh increments XDELTAX and XDELTAY are no more used**.
- LREAD_GROUND_PARAM : Flag to use or not the surface cover types (COVERnnn) and all other physiographic fields (except orographic ones) read in the PGD file.
  - .TRUE. to read the data in the PGD file
  - .FALSE. to use XUNIF_COVER idealized homogeneous values given in the namelist NAM_COVER (from the externalized surface) and scratch the PGD_FILE data
- LREAD_ZS : Flag to use or not the orography parameters read in the PGD file.
  - .TRUE. to use the data read in the PGD_FILE
  - .FALSE. to use an idealized orography given in the namelist NAM_GRIDH_PRE and scratch the PGD_FILE data

5.2.16 Namelist NAM_SLEVE (smoothed orography for Sleve coordinate)

Fortran name	Fortran type	default value
NSLEVE	integer	12
XSMOOTH_ZS	real	XUNDEF

- NSLEVE : number of iteration for computation of smooth orography.
- XSMOOTH_ZS : optional uniform smooth orography.

5.2.17 Namelist NAM_VER_GRID (contains vertical grid definition)

There are three ways to compute the vertical grid, as in PREP_REAL_CASE:

1. constant grid mesh: only the number of levels NKMAX and the grid mesh sizes ZDZGRD and ZDZTOP are used. ZDZGRD and ZDZTOP must have the same value. The type of grid YZGRID_TYPE is set to 'FUNCTN'.

2. two layers are defined, with constant stretching in each layer. The grid mesh size is given near the ground and at top of the model. It is possible that the top grid size is never reached, if the number of points is not enough for the prescribed stretchings. The type of grid YZGRID_TYPE is also set to 'FUNCTN'.

3. the levels are given by the user. The type of grid YZGRID_TYPE is set to 'MANUAL' in the namelist, and only the number of levels NKMAX is also used in it.

The variables of this namelist are:

Fortran name	Fortran type	default value
LTHINSHELL	logical	.FALSE.
NKMAX	integer	10
YZGRID_TYPE	6 characters	'FUNCTN'
ZDZGRD	real	300.
ZDZTOP	real	300.
ZZMAX_STRGRD	real	0.
ZSTRGRD	real	0.
ZSTRTOP	real	0.
LSLEVE	logical	FALSE
XLEN1	real	7500.
XLEN2	real	2500.

- LTHINSHELL : Flag for the thinshell approximation (logical)

- NKMAX : number of points in z-direction of the required physical domain. The total size of the array written in initial file will be $NKMAX + 2JPVEXT$ ($JPVEXT$ is fixed to 1 for the present version of Meso-NH)

- YZGRID_TYPE : type of vertical grid definition:

  - 'FUNCTN': the vertical grid is given by a regular logarithmic function, whose variation is determined by the values of free parameters ZDZGRD, ZDZTOP, ZSTRGRD, ZSTRTOP, ZZMAX_STRGRD described below.

  - 'MANUAL': the levels are explicitly given in the free-formatted part with the keyword `ZHAT` by entering the heights of the different levels from K=2 to K= KMAX + 2 (see 5.4.1).

- ZDZGRD : mesh length in z-direction near the ground

- ZDZTOP : mesh length in z-direction near the top of the model

- ZZMAX_STRGRD : Altitude separating the two constant stretching layers

- ZSTRGRD : Constant imposed stretching (in %) in the lower layer (below ZZMAX_STRGRD)

- ZSTRTOP : Constant imposed stretching (in %) in the upper layer (above ZZMAX_STRGRD)

- LSLEVE : flag for Sleve vertical coordinate.
- XLEN1 : decay scale for smooth topography (in meters)
- XLEN2 : decay scale for smale-scale topography deviation (in meters)

5.2.18 Namelist NAM_VPROF_PRE (variables for CIDEAL ='CSTN' or 'RSOU')

Fortran name	Fortran type	default value
LGEOSBAL	logical	FALSE
CFUNU	3 characters	ZZZ
CFUNV	3 characters	ZZZ
CTYPELOC	6 characters	IJGRID
XLATLOC	real	45.
XLONLOC	real	0.
XXHATLOC	real	20000.
XYHATLOC	real	20000.
NILOC	integer	4
NJLOC	integer	4

- LGEOSBAL : Flag to fulfill the geostrophic balance or not
  - .TRUE. the geostrophic balance is satisfied by the initial fields
  - .FALSE. the geostrophic balance is not satisfied by the initial fields
- CFUNU : String of 3 characters, describing the type of function, which gives the x component of the wind. Possible configurations are listed below
  - 'ZZZ' : U = U(z). The U(z) values are taken from the Radio-Sounding or analitycal profile given in the free-formatted part of the PRE_IDEA1.nam file.
  - 'Y*Z' : U= F(Y)*U(Z). The U(z) values are build in the same way as the 'ZZZ' case and the function F(Y) is a simple function of Y, which must be adapted by modifying its definition directly in the routine FUNUY. The default function is :

$$FUNUY(\hat{y}) = \frac{1}{\cosh\left(\frac{\hat{y}-\hat{y_0}}{zwidth}\right)}$$

– 'Y,Z' : U= G(Y,Z). The function G must also be adapted by modifying its definition directly in the routine FUNUYZ. The default function is :

$$FUNUYZ(\hat{y}, z) = \frac{1}{\cosh\left(\left(\frac{\hat{y}-\hat{y_0}}{zwidthy}\right)^2 + \left(\frac{z-z_0}{zwidthz}\right)^2\right)}$$

Notice that in this case the U(z) values given by the profile are not used.

- CFUNV : String of 3 characters, describing the type of function, which gives the y component of the wind. Possible configurations are listed below

  – 'ZZZ' : V = V(z). The V(z) values are taken from the Radio-Sounding or analitycal profile given in the free-formatted part of the PRE_IDEA1.nam file.

  – 'X*Z' : V= F(X)*V(Z). The V(z) values are build in the same way as the 'ZZZ' case and the function F(X) is a simple function of X, which must be adapted by modifying its definition directly in the routine FUNVX. The default function is :

$$FUNVX(\hat{x}) = \frac{1}{\cosh\left(\frac{\hat{x}-\hat{x_0}}{zwidth}\right)}$$

  – 'X,Z' : V= G(X,Z). The function G must also be adapted by modifying its definition directly in the routine FUNVXZ. The default function is :

$$FUNVXZ(\hat{x}, z) = \frac{1}{\cosh\left(\left(\frac{\hat{x}-\hat{x_0}}{zwidthx}\right)^2 + \left(\frac{z-z_0}{zwidthz}\right)^2\right)}$$

  Notice that in this case the V(z) values given by the profile are not used.

- CTYPELOC : Type of informations used to give the localization of vertical profile (string of 6 characters)

  – 'IJGRID' for (i,j) point on index space

  – 'XYHATM' for (x,y) coordinates on conformal plane or cartesian plane

  – 'LATLON' for (latitude,longitude) on spherical earth

- XLATLOC : Latitude (in degrees) of the vertical profile localization (used in case CTYPELOC='LATLON')

- XLONLOC : Longitude (in degrees) of the vertical profile localization (used in case CTYPELOC='LATLON')

- XXHATLOC : position (in meters) x of the vertical profile localization (used in cases CTYPELOC='XYHATM')

- XYHATLOC : position (in meters) y of the vertical profile localization (used in cases CTYPELOC='XYHATM')

- NILOC : position i of the vertical profile localization (used in cases CTYPELOC='IJGRID') If you use a 1D model, then NILOC is reset to 2 by the program.

- NJLOC : position j of the vertical profile localization (used in cases CTYPELOC='IJGRID') If you use a 1D or a 2D model, then NJLOC is reset to 2 by the program.

5.3 Namelists for the externalized surface

5.3.1 Principles

Further definition of surface parameters is not done by MESONH itself, but by the externalized surface included in it. Three cases are encountered:

1. You do not have any PGD input file or externalized surface. You have a fixed surface state (e.g., the surface temperature does not evolve) so you just need to set CSURF="NONE" in namelist NAM_GRn_PRE (default value).

2. You want to use the externalized surface (CSURF="EXTE" in namelist NAM_GRn_PRE) but you do not have any input PGD file or you do not want to use the surface fields included in it (LREAD_GROUND_PARAM = .FALSE.). Then, you must define both the physiographic and prognostic fields, and you must fill the following namelists (see SURFEX documentation for details):

   - NAM_PGD_SCHEMES
   - NAM_COVER
   - NAM_ISBA (if you chose to use the ISBA scheme).
   - NAM_CH_EMIS_PGD
   - NAM_DUMMY_PGD
   - NAM_PREP_SURF_ATM
   - NAM_PREP_SEAFLUX (if you chose to use the SEAFLX scheme)
   - NAM_PREP_WATFLUX (if you chose to use the WATFLX scheme)
   - NAM_PREP_TEB (if you chose to use the TEB urban scheme)
   - NAM_PREP_ISBA (if you chose to use the ISBA scheme)

   You can choose to :

(a) Use one or all the surface schemes. So you need to fill NAM_PGD_SCHEMES CSEA='SEAFLUX' or/and CNATURE='ISBA' or/and CWATER='WATFLUX' or/and CTOWN='TEB' and to set NAM_COVER XUNIF_COVER(i) where index "i" corresponds to the cover type, among those defined in routine mode_cover.f90 in SURFEX. Notice that no coherence check is performed between CSEA, CNATURE, CWATER, CTOWN on one side and the XUNIF_COVER type you choose on the other side. An example of namelist is given in the following part (Example 1).

(b) Prescribe your own surface fluxes and surface state. Before MASDEV49 version, you had to fill them in the dedicated routine init_ideal_flux.f90 in SURFEX and to recompile the routine. From MASDEV49, you just have to fill them in the SURFEX namelist of EXSEG1.nam (only for the run) : NAM_IDEAL_FLUX (see SURFEX user's guide). You need to fill also NAM_PGD_SCHEMES CSEA='FLUX' or/and CNATURE='FLUX' or/and CWATER='FLUX' or/and CTOWN='FLUX' / according to the surface type you consider and to set NAM_COVER XUNIF_COVER(i). NAM_PGD_SCHEMES CSEA='FLUX' and NAM_COVER XUNIF_COVER(1)=1. is often met. An example of namelist is given in the following part (Example 2).

3. You want to use all the informations contained in a PGD file. Only the prognostic variables must be defined, and the following namelists must be filled:

- NAM_PREP_SURF_ATM
- NAM_PREP_SEAFLUX (if you chose to use the SEAFLX scheme)
- NAM_PREP_WATFLUX (if you chose to use the WATFLX scheme)
- NAM_PREP_TEB (if you chose to use the TEB urban scheme)
- NAM_PREP_ISBA (if you chose to use the ISBA scheme)

An example of namelist is given in the following (Example 3).

Note that orography either comes from :

- the input PGD file (if any and if LREAD_ZS =.TRUE.). In this case, the atmospheric orography is also set equal to the one in this input PGD file.
- or from the orography you have defined from the MESONH namelists (in this case, the surface orography is forced to be equal to the atmosphere orography).

5.3.2 Examples :

Example 1 : You do not want to use a PGD file but you want to use a surface scheme, without prescribed fluxes :

```
&NAM_REAL_PGD /
&NAM_DIMn_PRE NIMAX=20, NJMAX=20 /
&NAM_VER_GRID NKMAX=36,YZGRID_TYPE='MANUAL'  /
&NAM_CONF_PRE LCARTESIAN=.TRUE., NVERB=10,
              CIDEAL='RSOU',  CZS='FLAT', LFORCING=.FALSE., LPACK=.FALSE.,
              LBOUSS=.FALSE., CEQNSYS='DUR',
              LPERTURB=.FALSE.  /
&NAM_PERT_PRE /
&NAM_CONFn LUSERV=.TRUE. /
&NAM_GRID_PRE XLAT0=35.762 /
&NAM_GRIDH_PRE XDELTAX=500.,XDELTAY=500.  /
&NAM_LUNITn CINIFILE='IDEA_ISBA' /
&NAM_PREP_ISBA XHUG_SURF=0., XHUG_ROOT=0.2, XHUG_DEEP=0.2,
               XTG_SURF=293., XTG_ROOT=293., XTG_DEEP=293. /
&NAM_POST_PRE /
&NAM_DYNn_PRE  /
&NAM_LBCn_PRE /
&NAM_VPROF_PRE LGEOSBAL = .FALSE., CTYPELOC='IJGRID', NILOC=2, NJLOC=2 /
&NAM_GRn_PRE CSURF='EXTE'/
&NAM_CH_MNHCn_PRE /
&NAM_BLANK /
&NAM_PGD_SCHEMES CNATURE='ISBA' /
&NAM_ISBA XUNIF_CLAY = 0.3, XUNIF_SAND = 0.3 /
&NAM_COVER XUNIF_COVER(208)=1. /
```

Example 2 : You do not want to use a PGD file and you want to prescribed your own fluxes (case ARM) :

```
&NAM_REAL_PGD /
&NAM_DIMn_PRE NIMAX=1, NJMAX=1 /
&NAM_CONF_PRE LCARTESIAN=.TRUE., NVERB=10,
              CIDEAL='RSOU',  CZS='FLAT', LFORCING=.TRUE., LPACK=.FALSE.,
              LBOUSS=.FALSE., CEQNSYS='DUR', LPERTURB=.FALSE.  /
&NAM_PERT_PRE /
```

```
&NAM_CONFn LUSERV=.TRUE. /
&NAM_GRID_PRE XLAT0=35.762 /
&NAM_GRIDH_PRE XDELTAX=40000., XDELTAY=40000. /
&NAM_VER_GRID  LTHINSHELL=.TRUE., NKMAX=100, ZDZGRD=40., ZDZTOP=40.,
               ZZMAX_STRGRD=1000. , ZSTRGRD=0., ZSTRTOP=0. /
&NAM_LUNITn CINIFILE='eurocs' /
&NAM_POST_PRE /
&NAM_DYNn_PRE  /
&NAM_LBCn_PRE CLBCX=2*"CYCL", CLBCY=2*"CYCL" /
&NAM_VPROF_PRE /
&NAM_GRn_PRE CSURF='EXTE'/
&NAM_CH_MNHCn_PRE /
&NAM_BLANK /
&NAM_PGD_SCHEMES CSEA='FLUX  ' /
&NAM_COVER XUNIF_COVER(1)=1. /
```

Example 3 : You want to use a PGD file and surface schemes :

```
&NAM_DIMn_PRE NIMAX=40, NJMAX=40 /
&NAM_VER_GRID NKMAX=36, YZGRID_TYPE='MANUAL' /
&NAM_CONF_PRE LCARTESIAN=.FALSE., CIDEAL='RSOU',LBOUSS=.FALSE., LPERTURB=.FALSE. ,
      CEQNSYS='DUR',NVERB=10 /
&NAM_GRn_PRE CSURF='EXTE' /
&NAM_REAL_PGD CPGD_FILE='PGD_CORSE', LREAD_ZS=.TRUE.,
              LREAD_GROUND_PARAM=.TRUE. /
&NAM_PGD_SCHEMES CNATURE='ISBA', CSEA='NONE',CWATER='WATFLX',CTOWN='TEB' /
&NAM_PREP_SURF_ATM NYEAR=2007, NMONTH=07, NDAY=26, XTIME=54000. /
&NAM_PREP_WATFLUX XTS_WATER_UNIF = 293. /
&NAM_PREP_ISBA XHUG_SURF=0.2, XHUG_ROOT=0.2, XHUG_DEEP=0.2, XTG_SURF=293.,
               XTG_ROOT=293., XTG_DEEP=293. /
&NAM_PREP_TEB XWS_ROAD=0., XWS_ROOF=0., XTS_ROAD=309.,
              XTS_ROOF= 298., XTS_WALL=298.,XTI_BLD=298., XTI_ROAD=298. /
&NAM_CONFn /
&NAM_GRIDH_PRE XDELTAX=250.,XDELTAY=250.  /
&NAM_LUNITn CINIFILE='IDEA_CORSE' /
&NAM_DYNn_PRE CPRESOPT= 'RICHA' NITR=10 XRELAX=1. /
&NAM_LBCn_PRE CLBCX= 2*'OPEN' CLBCY= 2*'OPEN' /
```

```
&NAM_VPROF_PRE CTYPELOC='IJGRID', NILOC=20, NJLOC=20,LGEOSBAL=.FALSE. /
```

5.4 Free-format part

Each section of the free format part must be introduced by its corresponding keyword (writen on a separated line)

There is always a moist variable written in PRE_IDEA1.nam file, even in idealized dry cases, for which the moist variable should be equal to zero in the PRE_IDEA1.nam file. The produced initial file will always contain a moist variable in 'CSTN' and 'RSOU' cases.

5.4.1 Optional Vertical grid :

keyword: **ZHAT**

If the vertical grid generation selector CZGRID_TYPE is equal to 'MANUAL', you must enter at the end of your namelist file, the heights of the vertical velocity levels. You must start from the ground level (K=2) to the model top (K=KMAX +2), thus you only have to enter KMAX + 1 values, because the level below the ground (i.e. K=1) is at the same distance from the ground (K=2) as the first level above the ground (K=3). Note also that the K= KMAX + 2 level represents the model top. In this case the free parameters (ZDZGRD, ZDZTOP, ZSTRGRD, ...) are not used

5.4.2 Radiosounding case :

keyword: **RSOU**

The radiosounding data are written in the free-format part of PRE_IDEA1.nam file, where the altitude variable is :

- the pressure in case KIND='STANDARD' or '**P**UVTHVMR' or '**P**UVTHVHU' or '**P**UVTHDHU' or '**P**UVTHDMR' (real, in Pascal)
- the height in case '**Z**UVTHVMR' or '**Z**UVTHVHU'or '**Z**UVTHDMR' or '**Z**UVTHLMR' (real, in meters)

The first wind variable is :

- the wind direction in case KIND='STANDARD' (real,in degrees)
- the zonal wind in cases KIND='P**UV**THVMR' or 'P**UV**THDMR' or 'Z**UV**THDMR' or 'Z**UV**THLMR'or 'Z**UV**THVHU' or 'P**UV**THDHU' or 'Z**UV**THVMR' or 'P**UV**THVHU' (real, in m/s)

The second wind variable is :

- the force direction in case KIND='STANDARD' (real, in m/s)

- the meridian wind in cases KIND='P**UV**THVMR' or 'P**UV**THDMR' or 'Z**UV**THDMR' or 'Z**UV**THLMR' or 'Z**UV**THVHU' or 'P**UV**THDHU' or 'Z**UV**THVMR' or 'P**UV**THVHU' (real, in m/s)

The temperature variable is :

- the temperature in case KIND='STANDARD' (real, in Kelvin)

- the virtual potential temperature in cases KIND='PUV**THV**MR' or 'PUV**THV**HU' or 'ZUV**THV**MR' or 'ZUV**THV**HU' (real,in Kelvin)

- the dry potential temperature in cases KIND='PUV**THD**MR' or 'PUV**THD**HU' or 'ZUV**THD**MR' (real, in Kelvin)

- the liquid potential temperature in case KIND='ZUV**THL**MR'(real, in Kelvin)

The moist variable is :

- the dew point temperature in case KIND='STANDARD' (real, in Kelvin)

- the vapor mixing ratio in cases KIND='PUVTHV**MR**' or 'ZUVTHD**MR**' or 'ZUVTHV**MR**' or 'PUVTHD**MR**' (real, in Kg/Kg)

- the total water mixing ratio in case KIND= 'ZUVTHLMR (real, in Kg/Kg)

- the relative humidity in cases KIND= 'ZUVTHV**HU**', or 'PUVTHD**HU**' or 'PUVTHV**HU**' (real, in percents)

Additional cloud variables

For the moment, this configuration works only for KIND='PUVTHDMR' or 'ZUVTHDMR' and L1D=.TRUE.. It is planned to compute radiation diagnostics with the **DIAG** program (see chapter 10).

- cloud mixing ratio if LUSERC=T or LUSERI=T (real, in Kg/Kg)

- ice mixing ratio if LUSERI=T (real, in Kg/Kg)

You should make sure that the levels are dense enough so that the Laplace relation, which gives the thickness between successive levels, can be applied. The radiosounding informations are written in the file in the following order :

- YEAR (integer, exemple : 1994), MONTH (integer, exemple : 4), DAY (integer, exemple : 22), TIME (real, in seconds, exemple : 36000 for 10 h)
- KIND of data used for the radiosounding (string of 8 charcaters) Nine kind of data are possible : 'STANDARD', 'PUVTHVMR', 'PUVTHVHU', 'ZUVTHVMR', 'ZUVTHVHU', 'PUVTHDMR', 'PUVTHDHU', 'ZUVTHDMR', 'ZUVTHLMR'.

  Except for the STANDARD kind :

  - the first letter of KIND represents the kind of altitude variable (P for pressure and Z for height),
  - the second and third letters represent the kind of wind variables (U for zonal wind, V for meridian wind),
  - the fourth, fifth and sixth letters represent the kind of temperature variable (THV for virtual potential temperature, THD for dry potential temperature and THL for liquid potential temperature),
  - the seventh and eighth letters represent the kind of moist variable (HU for relative humidity and MR for vapor mixing ratio).

  (In case of STANDARD kind, the altitude variable is the pressure, the wind variables are direction and wind force, the temperature variable is the temperature and the moist variable is the due point temperature.)
- HEIGHT of GROUND LEVEL (real, in meters)
- PRESSURE at GROUND LEVEL (real,in Pascal)
- a TEMPERATURE variable at GROUND LEVEL (real, in Kelvin)
- a MOIST variable at GROUND LEVEL
- NUMBER of WIND data LEVELS (integer)
- level 1 : ALTITUDE variable , first WIND variable, second WIND variable at wind level 1 (the lowest wind-level).
- level 2 : ALTITUDE variable, first WIND variable, second WIND variable.

$$\vdots$$

$$\vdots$$

- uppermost wind level : ALTITUDE variable, first WIND variable, second WIND variable.

- NUMBER of mass data LEVELS (integer) **Note that this number includes the ground level (i.e. the first level).** That is why the following list starts at level 2.

- level 2 : ALTITUDE variable, TEMPERATURE variable, MOIST variable, additional cloud variable(s) (the mass level 1 is at ground).

- level 3 : ALTITUDE variable, TEMPERATURE variable, MOIST variable, additional cloud variable(s) .

⋮

⋮

- uppermost mass level: ALTITUDE variable, TEMPERATURE variable, MOIST variable, additional cloud variable(s)

You should make sure that the highest level of the radiosounding is located above the highest vertical level of the model.

Example of free part of PRE_IDEA1.nam

```
RSOU
1990 10 3 72000.
'STANDARD'
200.
100240.
287.5
276.
2
85000. 20. 10.
70000. 30. 10.
3
90000. 280. 275.
60000. 271. 269.
```

5.4.3 Constant moist Brunt-Vaisala case :

keyword: **CSTN**

Data of the vertical profile are written in the free-format part of PRE_IDEA1.nam file in the following order :

- YEAR (integer, example : 1994), MONTH (integer, example : 4), DAY (integer, example : 22), TIME (real, in seconds, example : 36000. for 10 h)
- NUMBER of LEVELS (integer)
- VIRTUAL POTENTIAL TEMPERATURE at GROUND LEVEL (*i.e* at the first level) (real,in Kelvin)
- PRESSURE at GROUND LEVEL (*i.e* at the first level) (real, in Pascal)
- HEIGHT at all levels. **the first level is the ground level**
- ZONAL WIND COMPONENT at all levels (the first level is the ground level)
- MERIDIAN WIND COMPONENT at all levels (the first level is the ground level)
- RELATIVE HUMIDITY at all levels (the first level is the ground level)
- MOIST BRUNT VAISALA FREQUENCY at all layers (the number of layers is the number of levels - 1)

In this case, the level number can even be equal to 1, because the profile informations are linearly interpolated on the model grid without orography (wind components, θ_v and humidity) before the application of the Laplace relation to deduce the pressure and the vapor mixing ratio. Thus, the layers' thicknesses are never too large to invalidate the Laplace relation.

Example of free part of PRE_IDEA1.nam

```
CSTN
2006 06 06 21600.
5
287.5
100240.
200. 1000. 1500. 3000. 4000.
10. 20. 25. 30. 35.
2. 10. 12.5 11.5 15.
80. 84. 85. 79. 87.
0.01 0.014 0.015 0.016
```

5.4.4 The forced version

keyword: **ZFRC** or **PFRC**

For idealized simulations a forced mode can be useful to impose the effects of a simplified large scale environment to the model solution. This functionality works (LFORCING=.TRUE. in module MODD_CONF) when CIDEAL='RSOU' or 'CSTN' (see 5.2.10 and 5.3) and only in the case LCARTESIAN =.TRUE. and LGEOSBAL =.FALSE. for inclusion of a geostrophic wind forcing. All forcing fields are issued from spatial interpolation of chronological series of 1D data (provided by the user onto the model grid). They are prepared during the **prep_ideal_case** sequence and are stored in the LFI files for further use in case of RESTART model run.

The forcing fields can be time dependent. Application of the forcing begins as soon as the date and time of the first set of forcing field given by the user, is lower or equal to the current date and time of the model run. The forcing action of the last forcing field is remanant, this is a way to impose a stationnary forcing. When the current date and time of the model run is bounded by two successive forcing fields, a simple linear interpolation in time is made.

Note that an available Newtonian relaxation forcing type on $[u, v]$ and/or $[\theta, r_v]$ is exclusive from the other physical forcings.

The forcing informations and soundings have to be added at the end of the free-format part already written for CIDEAL='CSTN' or 'RSOU'. First, the type of forcing and the number of time dependent forcing are given:

- keyword forcing type (character*4)
  - ZFRC means that the altitude of the forcing data are in height scale (meters).
  - PFRC means that the altitude of the forcing data are in pressure scale (Pascal).
- number of time dependent forcing (integer)

The 1D forcing data are different from the one used to initialize the model because specific data have to be entered. The data used to define each forcing are given sequentially in the following order (one item per line):

- date and time of the forcing in the format:

  year (integer),

  month (integer),

  day (integer) and

  time of the day (real, s).
- ground height (real, m)
- ground pressure (real, Pa) (WARNING : in the MASDEV3_1 version either the ground height or the surface pressure was read, now we read both !)

- θ_d (real, K) at ground level (Nota: it is used later in the code to compute - if asked - a time varying sea surface temperature).
- r_v (real, kg/kg) at ground level
- number of level (integer)
- height of level1 (real, m) if ZFRC or pressure at level1 (real, Pa) if PFRC,

  u_{frc} component at level1 (real, m/s),

  v_{frc} component at level1 (real, m/s),

  θ_{frc} at level1 (real, K),

  $r_{v\ frc}$ at level1 (real, kg/kg),

  w_{frc} at level1 (real, m/s),

  $(\partial\theta/\partial t)_{frc}$ at level1 (real, K/s) and

  $(\partial r_v/\partial t)_{frc}$ at level1 (real, 1/s).

  $(\partial\theta/\partial x)_{frc}$ at level1 (real, K/m).

  $(\partial\theta/\partial y)_{frc}$ at level1 (real, K/m).
- idem at level2
- ...
- idem at levelN

If PFRC is the forcing type, an additional sounding is given in order to convert the pressure levels into height levels with enough accuracy. Data are organized as follows:

- number of level (integer)
- pressure at level1 (real, Pa),

  θ at level1 (real, K) and

  r_v at level1 (real, kg/kg).

This operation is repeated until the previous number of sounding is reached.

<u>Example of free part of PRE_IDEA1.nam</u>

```
ZFRC
   1
1983 07 01 0.
```

```
0
1000000
284.5
 .008
 6
   5.   -7.0   0.0  281.10   0.00540     -0.00000   0.0  0.0
  15.   -7.0   0.0  281.10   0.00540     -0.00000   0.0  0.0
1095.   -7.0   0.0  280.75   0.00540     -0.00300   0.0  0.0
1145.   -7.0   0.0  290.60   0.00190     -0.00300   0.0  0.0
3000.   -7.0   0.0  304.15   0.00190     -0.00300   0.0  0.0
9000.   -7.0   0.0  346.15   0.00190     -0.00300   0.0  0.0
```

5.4.5 Discretized orography

keyword: **ZSDATA**

Only the orography corresponding to the computational domain must be provided in the free format part. For 3D orography, data are read like if it was a map (the first line is the Northern border and the first data is the North-West corner) with one line per Y-axis increment.

Example of free part of PRE_IDEA1.nam

```
ZSDATA
30.    30.    35.     50.     30.     30.
30.    59.5  133.3   100.2   136.7   100.
35.    89.5  183.3   200.2   299.7   170.5
50.    112.5 193.0   210.2   206.7   120.
40.    82.5  153.0   180.5   156.7   100.3
```

5.5 Example of PRE_IDEA1.nam :

The selected case is the following:

- 2D mountain
- one moist layer atmosphere

FILE PRE_IDEA1.nam

```
&NAM_DIMn_PRE  NIMAX=128, NJMAX=1 /
&NAM_VER_GRID  NKMAX=32, YZGRID_TYPE = 'FUNCTN', ZDZGRD=500., ZDZTOP=500.,
               ZZMAX_STRGRD=1000.    , ZSTRGRD=0., ZSTRTOP= 0.,
```

```
&NAM_CONFn     LUSERV=.TRUE., NSV_USER = 0 /
&NAM_GRID_PRE  XLAT0 = 48.25 , XLON0 = 0.,
               XRPK  = 0.    , XBETA = 0.,
               XLONORI = 48.25, XLATORI = 0. /
&NAM_CONF_PRE  LCARTESIAN=.TRUE., LBOUSS=.FALSE.,
               CIDEAL='CSTN', CZS='BELL',
               LPERTURB= .FALSE., NVERB=1 /
&NAM_GRIDH_PRE XDELTAX=5.E2 , XDELTAY=5.E2,
               XHMAX=500., XAX=10.E3, XAY=10.E3, NIZS=64, NJZS=2,
               NEXPX = 1, NEXPY=1 /
&NAM_LUNITn    CINIFILE='HYD2D' /
&NAM_DYNn_PRE  CPRESOPT ='RICHA', NITR=4, XRELAX=1.0 /
&NAM_LBCn_PRE  CLBCX(1)='OPEN', CLBCX(2)='OPEN',
               CLBCY(1)='OPEN', CLBCY(2)='OPEN' /
&NAM_VPROF_PRE CTYPELOC='IJGRID', NILOC=10, NJLOC=2,
               CFUNU='ZZZ', CFUNV='ZZZ',
               LGEOSBAL=.FALSE. /
&NAM_GRn_PRE   CSURF='EXTE' /
&NAM_CH_MNHCn_PRE LUSECHEM = F /
CSTN
2
285.
100000.
0. 20000.
10. 10.
0. 0.
40. 40.
0.01
```

This file contains the informations necessary to generate the initial conditions for a quasi-hydrostatic flow, in the weakly non-linear regime, with a regular vertical grid.

Chapter 6

PREP_REAL_CASE

6.1 Presentation

PREP_REAL_CASE program performs the change of orography and vertical grid by interpolating horizontally and vertically for a GRIB file or only vertically for a MESO-NH file. The MESO-NH output file will be used either for the beginning of the simulation or for coupling. The main hypothesis is that hydrostatism is verified. Therefore, if the input file is a MESO-NH file, there is a small loss of information.

What's going in and out?

- Input:
  - a file containing the atmospheric 3D and surface 2D variable fields (hereafter called atmospheric file); it can be either
    - a GRIB file obtained from **extractecmwf** or **extractarpege**
    - or a MESO-NH file (obtained with **SPAWNING** for example)

    In the first case, both horizontal and vertical interpolations are carried utn by the PREP_REAL_CASE. In the second case, only vertical interpolation is carried on by PREP_REAL_CASE as the horizontal interpolation is already done by the spwaning (see chapter 7)
  - a physiographic data file (it can also be a complete MESO-NH file).
  - an optional file containing the chemical species (here after called chemical file); it is used only if the atmospheric file is a GRIB file. It can be either
    - a GRIB file obtained from **extractarpege** (e.g. an file from the Mocage french model)
    - or a MESO-NH file (obtained in a previous simulation for example)

 - the file PRE_REAL1.nam which contains the directives for PREP_REAL_CASE

- Output:

  - the MESO-NH FM-file

6.1.1 The physiographic data file

This is a FM file, but with fewer elements than a MESO-NH file. It contains the physiographic 2D fields. The geographic and grid data are stored on this file. This file is created by the program PREP_PGD. It is possible to use a complete MESO-NH file, since it also contains the physiographic fields. If one only wants to modify the vertical grid of a MESO-NH file, without any change to the orography, one can specify it as both atmospheric file and physiographic files.

It contains:

- the definition of the projection, the horizontal domain and the horizontal grid
- the physiographic fields

6.1.2 The atmospheric file

Both GRIB and FM files are self explanatory. The physiographic data stored in it will not be saved in the output MESONH file.

6.1.3 The surface file (optional)

Both GRIB and FM files are self explanatory. The surface fields can be read in another file than the atmospheric one.

6.1.4 The chemical file (optional)

Both GRIB and FM files are self explanatory. If the atmospheric file is a GRIB file, the chemical species can be read in another file than the atmospheric one.

6.2 The file PRE_REAL1.nam

This file contains namelists with the directives to run PREP_REAL_CASE. The namelists contain the names of the files and the definition of the vertical grid. The file can also contain a free formatted part after the vertical grid definition namelist, where the vertical levels can be prescribed if this option is chosen.

6.2.1 Namelist NAM_AERO_CONF (aerosol initialization)

Fortran name	Fortran type	default value
LORILAM	logical	FALSE
LDUST	logical	FALSE
LSALT	logical	FALSE
LINITPM	logical	FALSE
XINIRADIUSI	real	0.05
XINIRADIUSJ	real	0.2
XINISIGI	real	1.8
XINISIGJ	real	2.0
XN0IMIN	real	10.
XN0JMIN	real	1.
CRGUNIT	character	"NUMB"
NMODE_DST	integer	3
XN0MIN	real	1.e3 , 1.e1 , 1.e-2
XINIRADIUS	real	0.044, 0.3215, 1.575
XINISIG	real	2.0, 1.78, 1.85
CRGUNITD	character	"NUMB"
NMODE_SLT	integer	3
XN0MIN_SLT	real	1.e4 , 1.e2 , 1.e-1
XINIRADIUS_SLT	real	0.14, 1.125, 7.64
XINISIG_SLT	real	1.9, 2., 2.
CRGUNITS	character	"MASS"

See section 5.2.1 page 39 for details.

6.2.2 Namelist NAM_BLANK

See section 5.2.2 page 41 for details.

6.2.3 Namelist NAM_CONFZ

See section 5.2.6 page 44 for details.

6.2.4 Namelist NAM_FILE_NAMES (file names)

Fortran name	Fortran type	default value
HATMFILE	character (LEN=28)	' '
HATMFILETYPE	character (LEN=6)	'MESONH'
HPGDFILE	character (LEN=28)	' '
HSURFFILE	character (LEN=28)	' '
HSURFFILETYPE	character (LEN=6)	'MESONH'
HCHEMFILE	character (LEN=28)	' '
HCHEMFILETYPE	character (LEN=6)	'MESONH'
CINIFILE	character (LEN=28)	'INIFILE'

- HATMFILE : name of the atmospheric file.
- HATMFILETYPE : type of atmospheric file ('GRIBEX', 'MESONH')
- HPGDFILE : name of the Physiographic Data File.
- HSURFFILE : optional name of the file containing the surface fields.
- HSURFFILETYPE : type of surface file ('GRIBEX', 'MESONH')
- HCHEMFILE : optional name of the file containing the chemical species if they are not in the HATMFILE or if the ones of the HATMFILE are not used (only if HATMFILETYPE is 'GRIBEX'). The grids must be the same as the ones of the output file (CINIFILE).
- HCHEMFILETYPE : type of the chemical file ('GRIBEX', 'MESONH')
- CINIFILE : name of the MESO-NH output FM-file, used as initial or coupling file in a MESO-NH simulation

6.2.5 Namelist NAM_HURR_CONF (hurricane filtering and vortex bogussing)

Two PREP_REAL_CASE jobs are to be performed. In a mono-model configuration, the first job allows to remove analysed hurricane from the input GRIB fields: filtered and interpolated fields are written in a MesoNH file. It is used as input file for the second PREP_REAL_CASE job during which the analytical vortex is added.

Each step (hurricane filtering and vortex bogussing) is separately invoked within the PREP_REAL_CASE program:

i) The hurricane filtering is applied on four input atmospheric GRIB fields (HATMFILETYPE='GRIBEX'), when they are in the horizontal grid of the PGD file and in the vertical grid of the GRIB file. The input atmospheric GRIB fields filtered are the two horizontal components of wind, the absolute temperature and the surface pressure

reduced to ground level. Each field is decomposed into three parts: first, the BASic part is computed by the low-pass Barnes filter; then the hurricane (symmetric) disturbance is computed from the remainder disturbance part. The initial fields are then remplaced by their ENVironmental part: total field minus hurricane disturbance part.

ii) The vortex bogussing consists on a symmetric vortex added to the input atmospheric MesoNH fields (HATMFILETYPE='MESONH'). The tangential wind is computed from an analytical formulation (Holland, 1980): **Mercator projection must be used** to respect hypotheses of Holland. Then, the balanced mass field is deduced from the thermal wind relation. The bogus of the two horizontal components of wind and the potential temperature is added to the initial (filtered) fields.

For a grid-nesting simulation, the hurricane filtering is first applied for the outer domain (dad model), with the program PREP_REAL_CASE. The filtered fields are then horizontally interpolated for inner domains with the program SPAWNING (see section 7). Then, for each inner domain, a vortex bogussing is added with the program PREP_REAL_CASE.

Fortran name	Fortran type	default value
LFILTERING	logical	.FALSE.
CFILTERING	character (LEN=5)	'UVT '
NK	integer	50
XLAMBDA	real	0.2
XLATGUESS	real	XUNDEF
XLONGUESS	real	XUNDEF
XBOXWIND	real	XUNDEF
XRADGUESS	real	XUNDEF
NPHIL	integer	24
NDIAG_FILT	integer	-1
LBOGUSSING	logical	.FALSE.
XLATBOG	real	XUNDEF
XLONBOG	real	XUNDEF
XVTMAXSURF	real	XUNDEF
XRADWINDSURF	real	XUNDEF
XMAX	real	16000.
XC	real	0.7
XRHO_Z	real	-0.3
XRHO_ZZ	real	0.9
XB_0	real	1.65
XBETA_Z	real	-0.5
XBETA_ZZ	real	0.35
XANGCONV0	real	0.
XANGCONV1000	real	0.
XANGCONV2000	real	0.
CDADATMFILE	character (LEN=28)	' '
CDADBOGFILE	character (LEN=28)	' '

- LFILTERING : Flag to filter the fields (U,V,T,reduced Ps) of the atmospheric file (logical)

- CFILTERING : to choose the fields to be filtered (U,V,T,reduced Ps).
  - 'UVT ' : U,V,T are filtered (default),
  - 'UVTP ' : U,V,T and reduced PS are filtered,
- NK : number of points of the half-width of the window in which the Barnes filter is applied to compute low-pass component of a given field
- XLAMBDA : a coefficient in the exponential weighting function of the Barnes filter
- XLATGUESS : latitude of the guessed position of the cyclone center
- XLONGUESS : longitude of the guessed position of the cyclone center
- XBOXWIND : half-width of the box inside which the dynamical center is searched from the guessed position (km)
- XRADGUESS : guess of the radius of the domain in which the cyclone will be filtered (km)
- NPHIL : number of azimuthal directions used for the cylindrical coordinates
- NDIAG_FILT : allows storage of several components calculated from total fields. **Be careful, the components are on the GRIB vertical grid: in diaprog, plot them only on _K_ levels**. Then to visualize all the GRIB vertical levels, the number of MesoNH vertical levels must be equal or greater than the number of levels in the input GRIB file.

  0 : total (unfiltered) fields: UT15, VT15 for wind components; TEMPTOT, PRESTOT for absolute temperature and surface pressure,
  environmental (filtered) fields (total field minus hurricane disturbance component): UT16, VT16, TEMPENV, PRESENV,

  0,1 : basic fields (low-pass component isolated by the Barnes filter): UT17, VT17, TEMPBAS, PRESBAS,

  0,1,2 : total disturbance tangential wind component (XVTDIS).
- LBOGUSSING : Flag to switch on the addition of the bogus vortex (logical)
- XLATBOG : latitude of the bogussed position of the analytical cyclone center
- XLONBOG : longitude of the bogussed position of the analytical cyclone center

- XVTMAXSURF : maximum tangential wind near the surface or about 500 m altitude (m/s)
- XRADWINDSURF : radius of maximum wind near the surface or about 500 m altitude (km)
- XMAX : altitude where the tangential wind vanishes (m)
- The following variables are parameters describing tangential wind in Holland's law (see formulation in routine `holland_vt.f90`).

  XC : standard coefficient for maximum tangential wind,

  XRHO_Z, XRHO_Z : standard coefficients for radius of maximum wind,

  XB_0, XBETA_Z, XBETA_ZZ : standard coefficients for B parameter.
- XANGCONV0, XANGCONV1000, XANGCONV2000 : convergence angle of wind near the surface, at 1000m and 2000m altitude.
- CDADATMFILE : if LBOGUSSING=.TRUE. : name of the dad of HATMFILE
- CDADBOGFILE : if LBOGUSSING=.TRUE. : name of the dad of CINIFILE. The program will check that CDADATMFILE and CDADBOGFILE have the same characteristics, before replacing the dad name of CINIFILE by CDADBOGFILE instead of CDADATMFILE. CDADBOGFILE must exist before running the prep_real_case job.

6.2.6 Namelist NAM_REAL_CONF (configuration variables)

Fortran name	Fortran type	default value

CEQNSYS	character (LEN=3)	if HATMFILETYPE = 'GRIBEX':
		'DUR'
		if HATMFILETYPE = 'MESONH':
		CEQNSYS value used in input MESONH file

CPRESOPT	character (LEN=5)	'CRESI'
NVERB	integer	1

LSHIFT	logical	if HATMFILETYPE = 'GRIBEX':
		.TRUE.
		if HATMFILETYPE = 'MESONH':
		.FALSE.

LDUMMY_REAL	logical	.FALSE.
LRES	logical	.FALSE.
XRES	real	1.E-07

- CEQNSYS : EQuatioN SYStem
  - 'LHE': Lipps-HEmler 1982

  - 'MAE': Modified Anelastic Equations
  - 'DUR': following DURran 1990 derivations

- CPRESOPT : option for pressure solver ('RICHA', 'CGRAD', 'CRESI').
- NVERB : verbosity level (error diagnostics are computed if NVERB>4)
- LSHIFT : flag to shift altitudes in boundary layer
- LDUMMY_REAL : flag to read dummy fields stored in the GRIB file (if you have previously run **extractarpege** asking for additionnal fields by modifying FULLPOS namelist or **extractecmwf** by modifying MARS requests).You have to fill also a free formatted part as described in the exemple of AROME field in section 6.3 p.78).
- LRES : flag to change the residual divergence limit
- XRES : Value of the residual divergence limit

6.2.7 Namelist NAM_VER_GRID (vertical grid definition)

The use of the THINSHELL approximation is specified in this namelist. There are five ways to compute the vertical grid (the three first ones are similar to PREP_IDEAL_CASE):

1. constant grid mesh: only the number of levels NKMAX and the grid mesh sizes ZDZGRD and ZDZTOP are used. These must be equal. The type of grid YZGRID_TYPE is set to 'FUNCTN'.
2. two layers are defined, with constant stretching in each of these, the grid mesh sizes being given near the ground and at top of the model. It is possible that the top grid size is never reached, if the number of points is not enough for the prescribed stretchings. The type of grid YZGRID_TYPE is also set to 'FUNCTN'.
3. the levels are given by the user. The type of grid YZGRID_TYPE is set to 'MANUAL' in the namelist, and only the number of levels NKMAX is also used in it.
4. The levels in the output MESONH file are the same as in the input MESONH file (only if the atmospheric input file is a MESONH file). The type of grid YZGRID_TYPE is set to 'SAMEGR' (for "same grid") and NKMAX is not specified.
5. The physical levels of the output MESONH file are the same as the lower NKMAX physical levels in the input MESONH file (only if the atmospheric input file is a MESONH file). The type of grid YZGRID_TYPE is set to 'SAMEGR' (for "same grid") and NKMAX is specified.

The variables of this namelist are:

Fortran name	Fortran type	default value
LTHINSHELL	logical	.FALSE.
NKMAX	integer	60 if HATMFILETYPE='GRIBEX'
	integer	same as in input file if HATMFILETYPE='MESONH'
YZGRID_TYPE	character (len=6)	'FUNCTN' if HATMFILETYPE='GRIBEX'
		'SAMEGR' if HATMFILETYPE='MESONH'
ZDZGRD	real	300 m
ZDZTOP	real	300 m
ZZMAX_STRGRD	real	0 m
ZSTRGRD	real	0 %
ZSTRTOP	real	0 %
LSLEVE	logical	FALSE
XLEN1	real	7500.
XLEN2	real	2500.

- LTHINSHELL : Flag for the thinshell approximation (logical)
- NKMAX : number of points in z-direction of the required physical domain. The total size of the array written in initial file will be $NKMAX + 2JPVEXT$ ($JPVEXT$ is fixed to 1 for the present version of Meso-NH)
- YZGRID_TYPE : type of vertical grid definition:
  - 'FUNCTN' : the levels are calculated by the program, according to the namelist variables.
  - 'MANUAL' : the levels are written in the free-formatted part after the namelist.
  - 'SAMEGR' : the levels are the same as those in the input file. Only available when atmospheric input file is a MESONH file.
- ZDZGRD : mesh length in z-direction near the ground (m)
- ZDZTOP : mesh length in z-direction near the top of the model (m)
- ZZMAX_STRGRD : Altitude separating the two constant stretching layers (m)
- ZSTRGRD : Constant imposed stretching (in %) in the lower layer (below ZZMAX_STRGRD)
- ZSTRTOP : Constant imposed stretching (in %) in the upper layer (above ZZMAX_STRGRD)
- LSLEVE : flag for Sleve vertical coordinate.

- XLEN1 : decay scale for smooth topography (in meters)
- XLEN2 : decay scale for smale-scale topography deviation (in meters)

6.2.8 Namelists of the externalized surface for PREP_REAL_CASE

Fore more informations, see SURFEX documentation.

- NAM_PREP_SURF_ATM
- NAM_PREP_SEAFLUX
- NAM_PREP_WATFLUX
- NAM_PREP_FLAKE
- NAM_PREP_ISBA
- NAM_PREP_ISBA_SNOW
- NAM_PREP_ISBA_CARBON
- NAM_PREP_TEB
- NAM_PREP_TEB_SNOW
- NAM_PREP_TEB_GARDEN
- NAM_PREP_GARDEN_SNOW

6.2.9 Free formatted part : Vertical grid

This part is optional in the file, read only if YZGRID_TYPE='MANUAL'. It must begin by the keyword **ZHAT** In this case (NKMAX+1) levels are written in meters in free format after the keyword, from ground level (generally 0) to rigid top level.

6.2.10 Second free formatted part related to chemical species

: This part is only used if you have previously run **extractarpege** with MOCAGE outputs. This part must begin by the keyword **MOC2MESONH**. Then, the list of the MesoNH chemical species, and their corresponding GRIB code in the GRIB file, is specified as follows:

```
MOC2MESONH
transfer mocage/RACM variables (default values)
2 # NUMBER OF OPTIONAL GRIB VARIABLES
(A4,1X,I5)
O3   180
NO2  183
```

6.2.11 Examples of namelist file PRE_REAL1.nam

- GRIBEX file, levels being calculated, and chemical species

```
&NAM_FILE_NAMES HATMFILE   ='ALT90101500'     , HATMFILETYPE='GRIBEX' ,
                HPGDFILE   ='PGDFILE_10km'    ,
                CINIFILE   ='CPL_example1'    /
&NAM_REAL_CONF CEQNSYS="LHE", NVERB=7 /
&NAM_VER_GRID NKMAX=60, YZGRID_TYPE='FUNCTN', ZDZGRD=50., ZDZTOP=500.,
              ZZMAX_STRGRD=3000., ZSTRGRD=2., ZSTRTOP=6. /
&NAM_BLANK

MOC2MESONH
transfer mocage/RACM variables
2 # NUMBER OF OPTIONAL GRIB VARIABLES
(A4,1X,I5)
O3   180
NO2  183
```

N.B.: the mocage part is written at the end of namelist file.

- MESONH file and levels given manually

```
&NAM_FILE_NAMES HATMFILE   ='POI03.1.DAY01.001'   , HATMFILETYPE='MESONH' ,
                HPGDFILE   ='PGDFILE_10km'        ,
                CINIFILE   ='CPL_example2'        /
&NAM_REAL_CONF NVERB=5 /
&NAM_VER_GRID NKMAX=10, YZGRID_TYPE='MANUAL' /
ZHAT
0.
1050.
2100.
3250.
4300.
5200.
6100.
7000.
```

```
8000.
9000.
10000.
```

6.3 Processing of extra fields in AROME GRIB file

AROME GRIB files obtained with extractarpege contain fields which aren't read by default by PREP_REAL_CASE. You could want to have this extra fields in your FM file. To get this fields, you have to use LDUMMY_REAL=T and fill a free formatted part at the end of PRE_REAL1.nam.

This part must begin by the keyword **DUMMY_2D** (for 2D fields). Then, the list of the MesoNH dummy fields, and their corresponding GRIB code in the GRIB file (parameter code, and if ambiguity with another variables type and value of level), is specified as follows (This example matches with all the extra fields available in AROME GRIB file):

```
DUMMY_2D
diagnostiques prevus
10
ACCPLUIE 62
ACCNEIGE 79
ACCGRAUPEL 78
INSPLUIE 169
INSNEIGE 64
INSGRAUPEL 63
UM05 33 105 10
VM05 34 105 10
CLSHUMI.RELA 52 105 2
CLSTEMPE 11 105 2
```

Chapter 7

Horizontal interpolation from a MESO-NH file: SPAWNING

7.1 Presentation

This program performs the horizontal interpolation from one MESO-NH file into another (respectively file 1 and file 2). The grid of the file 2 must be exactly included in the grid of file 1. The file 2 can be directly used for a model run, but it contains smooth surface fields (especially the orography). It is possible to run the model with the two files with gridnesting interaction, since a iterative procedure insures the gridnesting condition on the orographies.

The domain of the file 2 can be defined either:

1. by namelist NAM_GRID2_SPA specification.
2. with the domain of another FM file, which grid is coherent with the input file. For example this file can be a PGD file created by **PREP_PGD** with a domain defined from the domain of file 1 and the same type of specifications as those in NAM_GRID2_SPA (see above).

7.2 The input SPAWN1.nam file

7.2.1 Namelist NAM_BLANK

See section 5.2.2 page 41 for details.

7.2.2 Namelist NAM_GRID2_SPA (manual definition of domain)

Fortran name	Fortran type	default value
IXOR	integer	1
IYOR	integer	1
IXSIZE	integer	file 1 domain
IYSIZE	integer	file 1 domain
IDXRATIO	integer	1
IDYRATIO	integer	1
GBAL_ONLY	logical	.FALSE.

- IXOR: first point I index, according to the file 1 grid, left to and out of the new physical domain.
- IYOR: first point J index, according to the file 1 grid, under and out of the new physical domain.
- IXSIZE: number of grid points in I direction, according to file 1 grid, recovered by the new domain. It must be equal to $2^m \times 3^n \times 5^p$ with $(m, n, p) \geq 0$
- IYSIZE: number of grid points in J direction, according to file 1 grid, recovered by the new domain. It must be equal to $2^m \times 3^n \times 5^p$ with $(m, n, p) \geq 0$
- IDXRATIO: resolution factor in I direction between the file 1 grid and the new grid. It must be equal to $2^m \times 3^n \times 5^p$ with $(m, n, p) \geq 0$
- IDYRATIO: resolution factor in J direction between the file 1 grid and the new grid. It must be equal to $2^m \times 3^n \times 5^p$ with $(m, n, p) \geq 0$
- GBAL_ONLY: Flag to enforce anelastic constraint only. The spawned file have the same characteristics as the CINIFILE one.

7.2.3 Namelist NAM_LUNIT2_SPA (file names)

Fortran name	Fortran type	default value
CINIFILE	character (len=28)	'INIFILE'
YDOMAIN	character (len=28)	none
YSPAFILE	character (len=28)	none
YSPANBR	character (len=2)	'00'
YDADINIFILE	character (len=28)	"
YDADSPAFILE	character (len=28)	"
YSONFILE	character (len=28)	"

- CINIFILE : name of the initial FM-file 1 (father domain) which will be used to spawn model 2.

- YDOMAIN : name of the file which defines the domain for model 2. If a domain file is provided for YDOMAIN, then all the information of namelist NAM_GRID2_SPA will be ignored.
- YSPAFILE : optional name of the spawned FM-file 2 (output file). If the user does not specify this name, or if YSPAFILE = CINIFILE, the code builds the spawned FM-file name as:

  YSPAFILE = CINIFILE.spaYSPANBR

  or YSPAFILE = CINIFILE.sprYSPANBR if YSONFILE is provided.
- YSPANBR : : NumBeR which will be added to CINIFILE to generate the FM-file name of the SPAwned file (string of 2 characters)
- YDADINIFILE : if GBAL_ONLY=.TRUE. : name of the CINIFILE dad.
- YDADSPAFILE : if GBAL_ONLY=.TRUE. : name of the YSPAFILE dad. Program will check that YDADINIFILE and YDADSPAFILE have the same characteristics, before replacing the dad name of YSPAFILE by YDADSPAFILE instead of YDADINIFILE. YDADSPAFILE must exist before running the spawning job.
- YSONFILE : optional name of a spawned FM-file (input file). It must have the same resolution as the spawned FM-file 2 (output file). The fields of YSONFILE will be used at points included in the domain defined by YDOMAIN or NAM_GRID2_SPA, instead of interpolated fields of CINIFILE. This allows to keep finest information when defining a new finest domain to follow atmospheric system.

7.2.4 Namelist NAM_SPAWN_SURF

Fortran name	Fortran type	default value
LSPAWN_SURF	logical	.TRUE.

- LSPAWN_SURF : : flag to perform or not the interpolation of all the surface fields (both physiographic and prognostic fields). Note that these interpolations are performed by the **externalized surface** facilities. However, no specific namelist is required for this operation.
  - LSPAWN_SURF = .TRUE. : the interpolations are made
  - LSPAWN_SURF = .FALSE. : the interpolations are not made and therefore no surface field will be present in the output spawned file.

Chapter 8

PREP_SURFEX

8.1 Presentation

PREP_SURFEX performs the interpolations of surface fields from one grid to another.

What's going in and out?

- Input:
  - a file containing the surface 2D variable fields (hereafter called input file); it can be either
    - a GRIB file obtained from **extractecmwf** or **extractarpege**
    - a MESO-NH file (obtained with **SPAWNING** for example)
  - a physiographic data file (it can also be a complete MESO-NH file).
  - the file PRE_REAL1.nam which contains the directives for PREP_SURFEX
- Output:
  - the FM-file containing the physiographic and pronostic surface fields.

8.2 The file PRE_REAL1.nam

This file contains namelists with the directives to run PREP_SURFEX. The namelists contain the names of the files.

1. Namelist NAM_FILE_NAMES: (contains file names)

Fortran name	Fortran type	default value
HATMFILE	character (LEN=28)	' '
HATMFILETYPE	character (LEN=6)	'GRIBEX'
HPGDFILE	character (LEN=28)	' '
CINIFILE	character (LEN=28)	'INIFILE'

- HATMFILE : name of the atmospheric file (up to 28 characters)
- HATMFILETYPE : type of the atmospheric file ('GRIBEX', 'MESONH')
- HPGDFILE : name of the Physiographic Data File (up to 28 characters)
- CINIFILE : name of the MESO-NH output FM-file, used as initial or coupling file in a MESO-NH simulation

2. externalized surface namelists for **PREP_SURFEX**

   The surface initial fields are produced by the **externalized surface** facilities. Refer to the **documentation of the surface** for more details. For **PREP_SURFEX**, you must fill the following namelists:

   - NAM_PREP_SURF_ATM
   - NAM_PREP_SEAFLUX (if you chose to use the SEAFLX scheme)
   - NAM_PREP_WATFLUX (if you chose to use the WATFLX scheme)
   - NAM_PREP_TEB (if you chose to use the TEB urban scheme)
   - NAM_PREP_ISBA (if you chose to use the ISBA scheme)

   CAUTION:

   (a) Note that all namelists can be void, but only if the initial file name for HATMFILE you provide in namelist NAM_FILE_NAMES contains the externalized surface fields.

   (b) If the file HATMFILE does not contain externalized surface fields, you must fill at least namelist NAM_PREP_SURF_ATM (if you want to initialize the surface prognostic fields from an input file). You can also define more precisely the surface fields by using the namelists for each scheme.

Chapter 9

Perform a MESONH simulation

9.1 Presentation

The MESONH user will specify some free parameters of the run by fixing their new values in the NAMELISTs of the file EXSEG$n.nam.

When more than one model is present, each model needs its own MESONH file to be initialized and its own EXSEG$n.nam file to fix the free-parameters (note that a lot of physical free-parameters depends on the mesh and therefore vary with the model number).

The input files are read by the program in order to realize the initialization and the eventual coupling of the MESONH model with a large-scale model (CEP, Arpège...).

The output files are of two types:

- synchronous files for a given instant of the run. They contain the prognostic fields and eventually, additionnal records for supplementary diagnostic fields at the same instant. The file name ends by `00n` with `n`>0

- a diachronic file for the temporal series of prognostic or diagnostic fields. The file name ends by `000`

9.2 The input EXSEG$n.nam file

We now describe in the following subsection the different NAMELISTs present in a complete EXSEG$n.nam file. Each variable present in a namelist of the EXSEG$n.nam file belongs to a declarative module whose name is related to the namelist name:

NAM_xxxxx $\Longrightarrow$ MODD_xxxxx

The documentation of the declarative modules MODD_xxxxx can be found in the Fortran code and contains a description of each variable of the Namelist NAM_xxxxx. Thus, we only

give the list of the subset of MODD_xxxxx present in the Namelist NAM_xxxxx with a short description of each parameter.

For instance, if no value is present for the variable CPRESOPT in the NAMELIST NAM_DYNn of EXSEG2.nam (the index 2 is for model 2), the model will take the value present in the MESONH file, used to initialize the model 2 for this segment. This information is present in the descriptive part of the MESONH file (see Chapter 3). If it is also absent from the MESONH initial file, the model will use its default value defined in the code.

9.2.1 Namelist NAM_ADVn (scalar advection schemes of model n)

Fortran name	Fortran type	default value
CUVW_ADV_SCHEME	6 characters	'CEN4TH'
CMET_ADV_SCHEME	6 characters	'PPM_01'
CSV_ADV_SCHEME	6 characters	'PPM_01'
NLITER	integer	2

It contains the different advection schemes for dynamic variables (u,v and w), scalar meteorological variables (temperature, water substances, TKE) and tracers used by the model n. They are included in the declarative module MODD_ADVn

- CUVW_ADV_SCHEME: Advection scheme used for horizontal and vertical velocities: The following options are possible :
  - 'CEN2ND' 2nd order advection scheme CENtred on space and time. It does NOT guarantee the sign preservation.
  - 'CEN4TH' 4th order advection scheme CENtred on space and time. It does NOT guarantee the sign preservation.
- CMET_ADV_SCHEME: Advection scheme used for the following METeorological variables: temperature, water substances and TKE. The following options are possible (see the Scientific Documentation for more details):
  - 'CEN2ND' 2nd order advection scheme CENtred on space and time. It does NOT guarantee the sign preservation.
  - 'CEN4TH' 4th order advection scheme CENtred on space and time. It does NOT guarantee the sign preservation.
  - 'FCT2ND' 2nd order advection scheme CENtred on space and time. It is POSITIVE definite.
  - 'MPDATA' 2nd order advection scheme uncentred on space and time. It is POSITIVE definite.

- 'PPM_00' PPM advection scheme without constraint
- 'PPM_01' Monotonic version of PPM. It is POSITIVE definite.

- CSV_ADV_SCHEME: Advection scheme used for the tracer variables. The same options as CMET_ADV_SCHEME can be used.

  Note that if LLG=T in NAM_CONF, CSV_ADV_SCHEME must be equal to CMET_ADV_SCHEME.

- NLITER : number of iterations that the MPDATA is applied. NLITER=1 (donor cell or upstream scheme).

9.2.2 Namelist NAM_BLANK (available variables)

See section 5.2.2 page 41 for details.

9.2.3 Namelist NAM_BUDGET (budget box description)

Fortran name	Fortran type	default value
CBUTYPE	4 characters	'NONE'
NBUMOD	integer	1
XBULEN	real	43200.
NBUKL	integer	1
NBUKH	integer	0
LBU_KCP	logical	TRUE
XBUWRI	real	43200.
NBUIL	integer	1
NBUIH	integer	0
NBUJL	integer	1
NBUJH	integer	0
LBU_ICP	logical	TRUE
LBU_JCP	logical	TRUE
NBU_MASK	integer	1

It contains the description of the box in which the budget are performed. This box is always built with a subset of points of the simulation box.

- CBUTYPE: type of box used to compute the budget:
  - 'CART' a cartesian box defined by the lowest and highest values of the indices in the 3 directions in the MESONH grid, defined in the following.
  - 'MASK' several areas, described by horizontal masks, are selected according to criteria evaluated at each model timestep. The budget computations are realized at the selected verticals for each criteria. The criteria must be defined in the routine *set_mask.f90*

- NBUMOD: number of the model in which the budget are performed. Only one model must be selected even if the grid-nesting is active.

- NBUMASK: Number of masks used to select the budgets' areas, in the case CBUTYPE= 'MASK'.

- XBULEN: Timesteo in seconds, on which the different source terms of all the budget are temporally averaged (the minimum value is 2$*$ XTSTEP).

- XBUWRI: Duration in seconds, between successive writings in the diachronic file of the budget storage arrays (CBUTYPE='MASK').

- NBUKL: value of the model level K for the bottom of the budget box, in the case of a cartesian box (CBUTYPE='CART').

- NBUKH: Same as NBUKL but for the top of the budget box. Inside the budget box:

$$NBUKL \leq K \leq NBUKH$$

- NBUJL: value of the model level J for the left side of the budget box, in the case of a cartesian box (CBUTYPE='CART').

- NBUJH: Same as NBUJL but for the right side of the budget box. Inside the budget box:

$$NBUJL \leq J \leq NBUJH$$

- NBUIL: value of the model level I for the left side of the budget box, in the case of a cartesian box (CBUTYPE='CART').

- NBUIH: Same as NBUIL but for the right side of the budget box. Inside the budget box:

$$NBUIL \leq J \leq NBUIH$$

- LBU_KCP: Flag to average or not in the K direction all the budget terms, for any CBUTYPE value.

- LBU_JCP: Flag to average or not in the J direction all the budget terms, for CBUTYPE='CART'.

- LBU_ICP: Flag to average or not in the I direction all the budget terms, for CBUTYPE='CART'.

The description of the budgets for every prognostic variable is given below. Because all the budgets are performed in the same way, we give here some details on the way to select or cumulate the different source terms.

Firstly, there is a flag to activate or not the budget of a given prognostic variable. It should be noted that the budget terms for the variable Ψ have the dimension of

$$\frac{\partial \left[\tilde{\rho}\Psi\right]}{\partial t}$$

Then, all the source terms computed in the model for this prognostic variable can be selected according to the following rules:

- NSOURCE_TERM= 0 if you do not want to take into account this source term(if you are only interested in some terms of a budget and not to the whole budget).
- NSOURCE_TERM= 1 if you select this source term as the first element of a set of source terms, which will be accumulated.
- NSOURCE_TERM= 2 if you select this source term and it is not the first element of a set of source terms, which are cumulated. Note that the source terms which come just after the last of this set of source term, must be discarded (NSOURCE_TERM= 0) or be the first element of a new cumul (NSOURCE_TERM= 1).

9.2.4 Namelist NAM_BU_RRC (budget for cloud water)

Fortran name	Meaning	Fortran type	default value
LBU_RRC	budget flag	logical	FALSE
NASSERC	time filter (Asselin)	integer	0
NNESTRC (if NMODEL> 1)	nesting	integer	0
NADVRC (if CMET_ADV_SCHEME='PPM_xx')	total advection	integer	0
NADVXRC (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along x	integer	0
NADVYRC (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along y	integer	0
NADVZRC (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along z	integer	0
NFRCRC (if LFORCING=T)	forcing	integer	0
NDIFRC (if LNUMDIFTH=T)	numerical diffusion	integer	0
NRELRC (if LHORELAX_RC=T or LVE_RELAX=T)	relaxation	integer	0

NDCONVRC (if CDCONV='KAFR' or CSCONV='KAFR')	KAFR convection	integer	0
NHTURBRC (if CTURB='TKEL' and CTURBDIM='3DIM')	horizontam turbulent diffusion	integer	0
NVTURBRC (if CTURB='TKEL')	vertical turbulent diffusion	integer	0
NNEGARC (if CCLOUD ≠ 'NONE')	negative correction	integer	0
NACCRRC (if CCLOUD≠ 'NONE' or 'REVE')	accretion	integer	0
NAUTORC (if CCLOUD≠ 'NONE' or 'REVE')	autoconversion into rain	integer	0
NSEDIRC (if (CCLOUD='C2R2' or 'KHKO' and LSEDC=T) or if (CCLOUD='ICEx' and LSEDIC=T))	sedimentation of cloud	integer	0
NCONDRC (if CCLOUD ≠ 'NONE' and 'ICEx')	vapor condensation or cloud water evaporation	integer	0
NHONRC (if CCLOUD='ICEx')	homogeneous nucleation	integer	0
NRIMRC (if CCLOUD='ICEx')	riming of cloud water	integer	0
NWETGRC (if CCLOUD='ICEx')	wet growth of graupel	integer	0
NDRYGRC (if CCLOUD='ICEx')	dry growth of graupel	integer	0
NIMLTRC (if CCLOUD='ICEx')	ice melting	integer	0
NBERFIRC (if CCLOUD='ICEx')	Bergeron-Findeisen gth.	integer	0
NCDEPIRC (if CCLOUD='ICEx')	condensation/deposition on ice	integer	0
NHENURC (if CCLOUD='C2R2' or 'KHKO')	CCN activation	integer	0
NSEDIRC (if CCLOUD='C2R2' or 'KHKO')	sedimentation	integer	0
NWETHRC (if CCLOUD='ICE4')	wet growth of hail	integer	0

9.2.5 Namelist NAM_BU_RRI (budget for non-precipitating ice)

Fortran name	Meaning	Fortran type	default value
LBU_RRI	budget flag	logical	FALSE
NASSERI	time filter (Asselin)	integer	0
NNESTRI (if NMODEL>1)	nesting	integer	0
NADVRI (if CMET_ADV_SCHEME='PPM_xx')	total advection	integer	0
NADVXRI (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along x	integer	0
NADVYRI (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along y	integer	0
NADVZRI (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along z	integer	0
NFRCRI (if LFORCING=T)	forcing	integer	0
NDIFRI (if LNUMDIFTH=T)	numerical diffusion	integer	0
NRELRI (if LHORELAX_RI=T or LVE_RELAX=T)	relaxation	integer	0
NDCONVRI (if CDCONV='KAFR' or CSCONV='KAFR')	KAFR convection	integer	0
NHTURBRI (if CTURB='TKEL' and CTURBDIM='3DIM')	horizontal turbulent diffusion	integer	0
NVTURBRI (if CTURB='TKEL')	vertical turbulent diffusion	integer	0
NNEGARI	negative correction	integer	0
NSEDIRI (if CCLOUD='ICEx')	sedimentation	integer	0
NHENURI (if CCLOUD='ICEx')	heterogenous nucleation	integer	0
NHONRI (if CCLOUD='ICEx')	homogeneous nucleation	integer	0
NAGGSRI (if CCLOUD='ICEx')	aggregation of snow	integer	0
NAUTSRI (if CCLOUD='ICEx')	autoconversion of ice	integer	0
NCFRZRI (if CCLOUD='ICEx')	conversion freezing	integer	0
NWETGRI (if CCLOUD='ICEx')	wet growth of graupel	integer	0
NDRYGRI (if CCLOUD='ICEx')	dry growth of graupel	integer	0

NIMLTRI (if CCLOUD='ICEx')	ice melting	integer	0
NBERFIRI (if CCLOUD='ICEx')	Bergeron-Findeisen gth.	integer	0
NCDEPIRI (if CCLOUD='ICEx')	condensation/deposition on ice	integer	0
NWETHRI (if CCLOUD='ICE4')	wet growth of hail	integer	0

9.2.6 Namelist NAM_BU_RRG (budget for graupel)

Fortran name	Meaning	Fortran type	default value
LBU_RRG	budget flag	logical	FALSE
NASSERG	time filter (Asselin)	integer	0
NNESTRG	nesting	integer	0
NADVRG (if CMET_ADV_SCHEME='PPM_xx')	total advection	integer	0
NADVXRG (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along x	integer	0
NADVYRG (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along y	integer	0
NADVZRG (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along z	integer	0
NFRCRG (if LFORCING=T)	forcing	integer	0
NDIFRG (if LNUMDIFTH=T)	numerical diffusion	integer	0
NRELRG (if LHORELAX_RG=T or LVE_RELAX=T)	relaxation	integer	0
NNEGARG	negative correction	integer	0
NSEDIRG	sedimentation	integer	0
NSFRRG	spontaneous freezing	integer	0
NDEPGRG	deposition on snow	integer	0
NRIMRG	riming of cloud water	integer	0
NCMELRG	conversion melting	integer	0
NCFRZRG	conversion freezing	integer	0
NWETGRG	wet growth of graupel	integer	0
NDRYGRG	dry growth of graupel	integer	0
NGMLTRG	graupel melting	integer	0
NWETHRG (if CCLOUD='ICE4')	wet growth of hail	integer	0
NACCRG	accretion of graupel	integer	0

9.2.7 Namelist NAM_BU_RRH (budget for hail)

Fortran name	Meaning	Fortran type	default value
LBU_RRH	budget flag	logical	FALSE
NASSERH	time filter (Asselin)	integer	0
NNESTRH	nesting	integer	0
NADVRH (if CMET_ADV_SCHEME='PPM_xx')	total advection	integer	0
NADVXRH (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along x	integer	0
NADVYRH (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along y	integer	0
NADVZRH (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along z	integer	0
NFRCRH (if LFORCING=T)	forcing	integer	0
NDIFRH (if LNUMDIFTH=T)	numerical diffusion	integer	0
NRELRH (if LHORELAX_RH=T or LVE_RELAX=T)	relaxation	integer	0
NNEGARH	negative correction	integer	0
NSEDIRH	sedimentation	integer	0
NWETGRH	wet growth of graupel	integer	0
NWETHRH	wet growth of hail	integer	0
NHMLTRH	hail melting	integer	0

9.2.8 Namelist NAM_BU_RRR (budget for rain water)

Fortran name	Meaning	Fortran type	default value
LBU_RRR	budget flag	logical	FALSE
NASSERR	time filter (Asselin)	integer	0
NNESTRR (if NMODEL> 1)	nesting	integer	0
NADVRR (if CMET_ADV_SCHEME='PPM_xx')	total advection	integer	0
NADVXRR (if CMET_ADV_SCHEME ≠ 'PPM_xx')	advection along x	integer	0
NADVYRR (if CMET_ADV_SCHEME ≠ 'PPM_xx')	advection along y	integer	0
NADVZRR (if CMET_ADV_SCHEME ≠ 'PPM_xx')	advection along z	integer	0
NFRCRR (if LFORCING=T)	forcing	integer	0
NDIFRR (if LNUMDIFTH=T)	numerical diffusion	integer	0
NRELRR (if LHORELAX_RR=T or LVE_RELAX=T)	relaxation	integer	0
NACCRRR	accretion of rain droplets	integer	0
NAUTORR	autoconversion into rain droplets	integer	0
NREVARR	rain evaporation	integer	0
NSEDIRR	sedimentation of rain droplets	integer	0
NSFRRR (if CCLOUD='ICEx')	spontaneous freezing	integer	0
NACCRR (if CCLOUD='ICEx')	accretion of rain water	integer	0
NCFRZRR (if CCLOUD='ICEx')	conversion freezing	integer	0
NWETGRR (if CCLOUD='ICEx')	wet growth of graupel	integer	0
NDRYGRR (if CCLOUD='ICEx')	dry growth of graupel	integer	0
NGMLTRR (if CCLOUD='ICEx')	graupel melting	integer	0
NWETHRR (if CCLOUD='ICEx')	wet growth of hail	integer	0
NHMLTRR (if CCLOUD='ICE4')	melting of hail	integer	0

9.2.9 Namelist NAM_BU_RRS (budget for snow)

Fortran name	Meaning	Fortran type	default value
LBU_RRS	budget flag	logical	FALSE
NASSERS	time filter (Asselin)	integer	0
NNESTRS (if NMODEL> 1)	nesting	integer	0
NADVRS (if CMET_ADV_SCHEME='PPM_xx')	total advection	integer	0
NADVXRS (if CMET_ADV_SCHEME ≠ 'PPM_xx')	advection along x	integer	0
NADVYRS (if CMET_ADV_SCHEME ≠ 'PPM_xx')	advection along y	integer	0
NADVZRS (if CMET_ADV_SCHEME ≠ 'PPM_xx')	advection along z	integer	0
NFRCRS (if LFORCING=T)	forcing	integer	0
NDIFRS (if LNUMDIFTH=T)	numerical diffusion	integer	0
NRELRS (if LHORELAX_RS=T or LVE_RELAX=T)	relaxation	integer	0
NNEGARS	negative correction	integer	0
NSEDIRS (if CCLOUD='ICEx')	sedimentation	integer	0
NDEPSRS (if CCLOUD='ICEx')	deposition on snow	integer	0
NAGGSRS (if CCLOUD='ICEx')	aggregation of snow	integer	0
NAUTSRS (if CCLOUD='ICEx')	autoconversion of ice	integer	0
NRIMRS (if CCLOUD='ICEx')	riming of cloudwater	integer	0
NACCRS (if CCLOUD='ICEx')	accretion of rainwater	integer	0
NCMELRS (if CCLOUD='ICEx')	conversion melting	integer	0
NWETGRS (if CCLOUD='ICEx')	wet growth of graupel	integer	0
NDRYGRS (if CCLOUD='ICEx')	dry growth of graupel	integer	0
NWETHRS (if CCLOUD='ICE4')	wet growth of hail	integer	0

9.2.10 Namelist NAM_BU_RRV (budget for vapor)

Fortran name	Meaning	Fortran type	default value
LBU_RRV	budget flag	logical	FALSE
NASSERV	time filter (Asselin)	integer	0
NNESTRV (if NMODEL> 1)	nesting	integer	0
NADVRV (if CMET_ADV_SCHEME='PPM_xx')	total advection	integer	0
NADVXRV (if CMET_ADV_SCHEME ≠ 'PPM_xx')	advection along x	integer	0
NADVYRV (if CMET_ADV_SCHEME ≠ 'PPM_xx')	advection along y	integer	0
NADVZRV (if CMET_ADV_SCHEME ≠ 'PPM_xx')	advection along z	integer	0
NFRCRV (if LFORCING=T)	forcing	integer	0
NNUDRV (if LNUDGING=T)	nudging	integer	0
NDIFRV (if LNUMDIFTH=T)	numerical diffusion	integer	0
NRELRV (if LHORELAX_RV=T or LVE_RELAX=T)	relaxation	integer	0
NDCONVRV (if CDCONV='KAFR' or CSCONV='KAFR')	KAFR convection	integer	0
NMAFLRV (if CSCONV='EDKF')	Mass flux	integer	0
NHTURBRV (if CTURB='TKEL' and CTURBDIM='3DIM')	horizontal turbulent diffusion	integer	0
NVTURBRV (if CTURB='TKEL')	vertical turbulent diffusion	integer	0
NNEGARV (if CCLOUD ≠ 'NONE')	negative	integer	0
NREVARV (if CCLOUD≠ 'NONE' or 'REVE')	rain evaporation	integer	0
NCONDRV	vapor condensation or cloud water evaporation	integer	0
NHENURV (if CCLOUD='ICE3','ICE4','KHKO','C2R2')	heterogenous nucleation	integer	0
NDEPSRV (if CCLOUD='ICE3' or 'ICE4')	deposition on snow	integer	0
NDEPGRV (if CCLOUD='ICE3' or 'ICE4')	deposition on graupel	integer	0
NCDEPIRV (if CCLOUD='ICE3' or 'ICE4')	condensation/deposition on ice	integer	0

9.2.11 Namelist NAM_BU_RSV (budget for a Scalar Variable)

Fortran name	Meaning	Fortran type	default value
LBU_RSV	budget flag	logical	FALSE
NASSESV	time filter (Asselin)	integer	0
NNESTSV	nesting	integer	0
NADVSV (if CSV_ADV_SCHEME='PPM_xx')	total advection	integer	0
NADVXSV (if CSV_ADV_SCHEME ≠ 'PPM_xx')	advection along x	integer	0
NADVYSV (if CSV_ADV_SCHEME ≠ 'PPM_xx')	advection along y	integer	0
NADVZSV (if CSV_ADV_SCHEME ≠ 'PPM_xx')	advection along z	integer	0
NFRCSV (if LFORCING=T)	forcing	integer	0
NDIFSV (if LNUMDIFSV=T)	numerical diffusion	integer	0
NRELSV (if LHORELAX_SV=T or LVE_RELAX=T)	relaxation	integer	0
NDCONVSV (if CDCONV='KAFR' or CSCONV='KAFR')	KAFR convection	integer	0
NMAFLSV (if CSCONV='EDKF')	Mass flux	integer	0
NHTURBSV (if CTURB='TKEL' and CTURB-DIM='3DIM')	horizontal turbulent diffusion	integer	0
NVTURBSV (if CTURB='TKEL')	vertical turbulent diffusion	integer	0
NCHEMSV	chemistry activity	integer	0

9.2.12 Namelist NAM_BU_RTKE (budget for TKE)

Fortran name	Meaning	Fortran type	default value
LBU_RTKE	budget flag	logical	FALSE
NASSETKE	time filter (Asselin)	integer	0
NADVTKE (if CMET_ADV_SCHEME='PPM_xx')	total advection	integer	0
NADVXTKE (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along x	integer	0
NADVYTKE (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along y	integer	0
NADVZTKE (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along z	integer	0
NFRCTKE (if LFORCING=T)	forcing	integer	0
NDIFTKE (if LNUMDIFTH=T)	numerical diffusion	integer	0
NRELTKE (if LHORELAX_TKE=T or LVE_RELAX=T)	relaxation	integer	0
NDPTKE	dynamic production	integer	0
NTPTKE	thermal production	integer	0
NDISSTKE	dissipation of TKE	integer	0
NTRTKE	turbulent transport	integer	0

9.2.13 Namelist NAM_BU_RTH (budget for TH)

Fortran name	Meaning	Fortran type	default value
LBU_RTH	budget flag	logical	FALSE
NASSETH	time filter (Asselin)	integer	0
NNESTTH (if NMODEL> 1)	nesting	integer	0
NADVTH (if CMET_ADV_SCHEME='PPM_xx')	total advection	integer	0
NADVXTH (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along x	integer	0
NADVYTH (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along y	integer	0
NADVZTH (if CMET_ADV_SCHEME $\neq$ 'PPM_xx')	advection along z	integer	0
NFRCTH (if LFORCING=T)	forcing	integer	0
NNUDTH (if LNUDGING=T)	nudging	integer	0
NPREFTH (if LHORELAX_UVWTH=T or LVE_RELAX=T)	reference pressure term	integer	0

NDIFTH (if LNUMDIFTH=T)	numerical diffusion	integer	0
NRELTH (if LHORELAX_UVWTH=T or LVE_RELAX=T)	relaxation	integer	0
NRADTH (if CRAD $\neq$ 'NONE')	radiation	integer	0
NDCONVTH (if CDCONV='KAFR' ou CSCONV='KAFR')	KAFR convection	integer	0
NMAFLTH (if CSCONV='EDKF')	Mass flux	integer	0
NHTURBTH (if CTURB='TKEL' and CTURB-DIM='3DIM')	horizontal turbulent diffusion	integer	0
NVTURBTH (if CTURB='TKEL')	vertical turbulent diffusion	integer	0
NDISSHTH (if CTURB='TKEL')	dissipation	integer	0
NNEGATH (if CCLOUD $\neq$ 'NONE')	negative	integer	0
NREVATH (if rain is activated)	rain evaporation	integer	0
NCONDTH	vapor condensation or cloud water evaporation	integer	0
NHENUTH (if CCLOUD='ICE3','ICE4','KHKO','C2R2')	heterogeneous nucleation	integer	0
NHONTH (if CCLOUD='ICE3' or 'ICE4')	homogeneous nucleation	integer	0
NSFRTH (if CCLOUD='ICE3' or 'ICE4')	spontaneous freezing	integer	0
NDEPSTH (if CCLOUD='ICE3' or 'ICE4')	deposition of snow	integer	0
NDEPGTH (if CCLOUD='ICE3' or 'ICE4')	deposition of graupel	integer	0
NRIMTH (if CCLOUD='ICE3' or 'ICE4')	riming of cloud	integer	0
NACCTH (if CCLOUD='ICE3' or 'ICE4')	accretion of rain	integer	0
NCFRZTH (if CCLOUD='ICE3' or 'ICE4')	conversion freezing	integer	0
NWETGTH (if CCLOUD='ICE3' or 'ICE4')	wet growth of graupel	integer	0
NDRYGTH (if CCLOUD='ICE3' or 'ICE4')	dry growth of graupel	integer	0
NGMLTTH (if CCLOUD='ICE3' or 'ICE4')	graupel melting	integer	0
NIMLTTH (if CCLOUD='ICE3' or 'ICE4')	ice melting	integer	0
NBERFITH (if CCLOUD='ICE3' or 'ICE4')	bergeron-findeisen	integer	0
NCDEPITH (if CCLOUD='ICE3' or 'ICE4')	condensation/deposition on ice	integer	0
NWETHTH (if CCLOUD='ICE4')	wet growth of hail	integer	0
NHMLTTH (if CCLOUD='ICE4')	melting of hail	integer	0

9.2.14 Namelist NAM_BU_RU (budget for U)

LBU_RU	budget flag	logical	FALSE
NASSEU	time filter (Asselin)	integer	0
NNESTU (if NMODEL> 1)	nesting	integer	0
NADVXU	advection along x	integer	0
NADVYU	advection along y	integer	0
NADVZU	advection along z	integer	0
NFRCU (if LFORCING=T)	forcing	integer	0
NNUDU (if LNUDGING=T)	nudging	integer	0
NCURVU (if LCARTESIAN=F)	curvature terms	integer	0
NCORU (if LCORIO=T)	Coriolis term	integer	0
NDIFU (if LNUMDIFU=T)	numerical diffusion	integer	0
NRELU (if LHORELAX_UVWTH=T or LVE_RELAX=T)	relaxation	integer	0
NVTURBU (if CTURB='TKEL')	vertical turbulent diffusion	integer	0
NHTURBU (if CTURB='TKEL' and CTURBDIM='3DIM')	horizontal turbulent diffusion	integer	0
NPRESU	pressure term	integer	0
NMAFLU (if CSCONV='EDKF')	mass flux	integer	0

9.2.15 Namelist NAM_BU_RV (budget for V)

LBU_RV	budget flag	logical	FALSE
NASSEV	time filter (Asselin)	integer	0
NNESTV (if NMODEL> 1)	nesting	integer	0
NADVXV	advection along x	integer	0
NADVYV	advection along y	integer	0
NADVZV	advection along z	integer	0
NFRCV (if LFORCING=T)	forcing	integer	0
NNUDV (if LNUDGING=T)	nudging	integer	0
NCURVV (if LCARTESIAN=F)	curvature terms	integer	0
NCORV (if LCORIO=T)	Coriolis term	integer	0
NDIFV (if LNUMDIFU=T)	numerical diffusion	integer	0
NRELV (if LHORELAX_UVWTH=T or LVE_RELAX=T)	relaxation	integer	0
NHTURBV (if CTURB='TKEL' and CTURBDIM='3DIM')	horizontal turbulent diffusion	integer	0
NVTURBV (if CTURB='TKEL')	vertical turbulent diffusion	integer	0
NPRESV	pressure term	integer	0
NMAFLV (if CSCONV='EDKF')	mass flux	integer	0

9.2.16 Namelist NAM_BU_RW (budget for W)

LBU_RW	budget flag	logical	FALSE
NASSEW	time filter (Asselin)	integer	0
NNESTW (if NMODEL> 1)	nesting	integer	0
NADVXW	advection along x	integer	0
NADVYW	advection along y	integer	0
NADVZW	advection along z	integer	0
NFRCW (if LFORCING=T)	forcing	integer	0
NNUDW (if LNUDGING=T)	nudging	integer	0
NCURVW (if LCARTESIAN=F)	curvature terms	integer	0
NCORW (if LCORIO=T)	Coriolis term	integer	0
NGRAVW	gravity term	integer	0
NDIFW (if LNUMDIFU=T)	numerical diffusion	integer	0
NRELW (if LHORELAX_UVWTH=T or LVE_RELAX=T)	relaxation	integer	0
NHTURBW (if CTURB='TKEL' and CTURBDIM='3DIM')	horizontal turbulent diffusion	integer	0
NVTURBW (if CTURB='TKEL')	vertical turbulent diffusion	integer	0
NPRESW	pressure term	integer	0

9.2.17 Namelist NAM_CH_MNHCn (control of MNHC)

Fortran name	Fortran type	default value
LUSECHEM	logical	FALSE
LUSECHAQ	logical	FALSE
LUSECHIC	logical	FALSE
LCH_INIT_FIELD	logical	FALSE
LCH_SURFACE_FLUX	logical	FALSE
LCH_CONV_SCAV	logical	FALSE
LCH_CONV_LINOX	logical	FALSE
LCH_PH	logical	FALSE
LCH_RET_ICE	logical	FLASE
XCH_PHINIT	real	5.2
XRTMIN_AQ	real	5.e-8
CCHEM_INPUT_FILE	80 characters	'EXSEG1.nam'
CCH_TDISCRETIZATION	10 characters	'SPLIT'
NCH_SUBSTEPS	integer	1
LCH_TUV_ONLINE	logical	TRUE
CCH_TUV_LOOKUP	80 characters	'PHOTO.TUV39'
CCH_TUV_CLOUDS	4 characters	'NONE'
XCH_TUV_ALBNEW	real	-1.
XCH_TUV_DOBNEW	real	-1.
XCH_TUV_TUPDATE	real	600.
CCH_VEC_METHOD	3 characters	'MAX'
NCH_VEC_LENGTH	integer	1000
XCH_TS1D_TSTEP	real	600.
CCH_TS1D_COMMENT	80 characters	'no comment'
CCH_TS1D_FILENAME	80 characters	'IO1D'

- LUSECHEM: switch to activate chemistry.
- LUSECHAQ: switch to activate aqueous phase chemistry.
- LUSECHIC: switch to activate ice phase chemistry. This means that several pronostics variables are added equal to the number of solubles gases. These variables represent the mixing ratio of the soluble gases inside the precipitating iced hydrometeors.
- LCH_INIT_FIELD: switch to activate initialization subroutine CH_INIT_FIELD_n.
- LCH_SURFACE_FLUX: switch to activate chemical surface fluxes i.e. emissions, LCH_SURF_EMIS must be set to TRUE in namelist NAM_CH_SURFn of SURFEX.
- LCH_CONV_SCAV: switch to activate scavenging of chemical species (gazeous or aerosol) and dusts by convective precipitations.
- LCH_CONV_LINOX: switch to activate the production of NOx by LIghtning flashes inside deep convective clouds and its transport (LCHTRANS must be set to TRUE).

  - LUSECHEM=.F. : a scalar variable named LINOX are written in the LFI file
  - LUSECHEM=.T. : the convective source is added to the NO chemical variable.

- LCH_PH: switch to activate the computing of pH in cloud water and rainwater as diagnostic variables. XPHC and XPHR are added in LFI output files.
- LCH_RET_ICE: switch to activate the retention of solubles gase in iced hydrometeors without considering additional pronostics variables. LUSECHIC is set to FALSE. Be carefull this option leads to a loss of mass.
- XCH_PHINIT: pH value when aqueous phase chemistry is activated (LUSECHAQ is set to TRUE).
  - LCH_PH=.T. : XCH_PHINIT is the initial pH value,
  - LCH_PH=.F. : XCH_PHINIT is the constant pH value during the whole simulation.
- XRTMIN_AQ: when aqueous phase chemistry is activated (LUSECHAQ is set to TRUE), XRTMIN_AQ is the threshold value for cloud water (or rainwater) content from which aqueous phase chemistry and exchange between gas and liquid phases are computed.
- CCHEM_INPUT_FILE: name of the general purpose input file.
- CCH_TDISCRETIZATION: temporal discretization
  - CCH_TDISCRETIZATION='SPLIT': use time-splitting, input fields for solver are scalar variables at t+dt (derived from XRSVS)
  - CCH_TDISCRETIZATION='CENTER': use centered tendencies, input fields for solver are scalar variables at t (XSVT)
  - CCH _TDISCRETIZATION='LAGGED': use lagged tendencies, input fields for solver are scalar variables at t-dt (XSVM)
- NCH_SUBSTEPS: number of steps to be taken by the solver during two time steps of MesoNH; the time step of the solver is thus equal to 2*XTSTEP/NCH_SUBSTEPS
- LCH_TUV_ONLINE: switch to activate online photolysis rates calculations (only for 1D simulation). If false, photolysis rates are pre-calculated as a function of solar zenith angle and surface albedo and interpolated on the model grid.
- CCH_TUV_LOOKUP: name of the lookup table file.
- CCH_TUV_CLOUDS: method for calculating the impact of clouds on UV radiations (only for 3-D version)

  - CCH_TUV_CLOUDS='NONE' : No cloud correction on UV radiations
  - CCH_TUV_CLOUDS='CHAN' : Cloud correction on UV radiations following Chang et al., [1987]

- XCH_TUV_ALBNEW: surface albedo for photolysis rates calculations (only for 1-D version. For 3-D version, albedos are prescribed as a function of the surface characteristics).
- XCH_TUV_DOBNEW: scaling factor for ozone column dobson.
- XCH_TUV_TUPDATE: update frequency to refresh photolysis rates.
- CCH_VEC_METHOD: type of vectorization mask
  - 'MAX' take NCH_VEC_LENGTH points
  - 'TOT' take all grid points
  - 'HOR' take horizontal layers
  - 'VER' take vertical columns

- NCH_VEC_LENGTH: number of points for 'MAX' option.
- XCH_TS1D_TSTEP: time between two call to write_ts1d.
- CCH_TS1D_COMMENT: comment for write_ts1d.
- CCH_TS1D_FILENAME: filename for write_ts1d files.

9.2.18 Namelist NAM_CH_ORILAM

This namelist is used to activate ORILAM chemical aerosols (lognormal distribution for Aitken and accumulation mode). This parameterization include coagulation (intra and inter modal), nucleation, sedimentation, condensation/adsorption of gas phase. This parameterization need to be run together with gas chemical phase (namelist NAM_CH_MNHCn). For correct representation, it is recommended to have severals compounds as HNO3 (nitric acid), H2SO4 (or SULF; sulphates), NH3 (ammonium) and CO (carbon monoxyde) in the chemical scheme.

Fortran name	Fortran type	default value
LORILAM	logical	FALSE
LVARSIGI	logical	FALSE
LVARSIGJ	logical	FALSE
LSEDIMAERO	logical	FALSE
XINIRADIUSI	real	0.01
XINIRADIUSJ	real	0.5
CRGUNIT	character (len=4)	'MASS'
XINISIGI	real	1.60
XINISIGJ	real	1.60
XN0IMIN	real	10.
XN0JMIN	real	1.
XCOEFRADIMAX	real	10.
XCOEFRADJMAX	real	10.
XCOEFRADIMIN	real	.1
XCOEFRADJMIN	real	.1
CMINERAL	character (len=5)	'NONE'
CORGANIC	character (len=5)	'NONE'
CNUCLEATION	character (len=80)	'NONE'

- LORILAM: flag to activate chemical aerosol (only if LUSECHEM = .TRUE.).
- LVARSIGI: flag to activate variable standard deviation for mode I (Aitken).
- LVARSIGJ: flag to activate variable standard deviation for mode J (accumulation).
- LSEDIMAERO: flag to activate aerosol sedimentation.
- XINIRADIUSI: flag for the initialization of mean radius mode I (Aitken mode) of the distribution (in micrometers).
- XINIRADIUSJ: flag for the initialization of mean radius mode J (accumulation mode) of the distribution (in micrometers).
- CRGUNIT: type of mean radius given in namelist. Default is for a mass spectral distribution; XINIRADIUSI and XINIRADIUSJ have been converted into a mean radius in number.
  IF CRGUNIT $\neq$ 'MASS' then the mean radius need to be given for a number spectral distribution (no conversion).
- XINISIGI: value of standard deviation for mode I (Aitken mode).
- XINISIGJ: value of standard deviation for mode J (accumulation mode).
- XCOEFRADIMAX: factor to compute maximum value of mean radius mode I (Aitken mode). $R_i^{max} = XCOEFRADIMAX.XINIRADIUSI$

- XCOEFRADJMAX: factor to compute maximum value of mean radius mode J (accumulation mode). $R_j^{max} = XCOEFRADJMAX.XINIRADIUSJ$

- XCOEFRADIMIN: same as XCOEFRADIMAX but for the minimum value.

- XCOEFRADJMIN: same as XCOEFRADIMAX but for the minimum value.

- CMINERAL: type of parameterization for mineral gas/particle balance. Possible values are:
  CMINERAL = 'ARES' : ARES parameterization (non vectorized)
  CMINERAL = 'NARES': neuronal network of ARES (vectorized)
  CMINERAL = 'ISPIA': ISORROPIA parameterization (non vectorized)
  CMINERAL = 'TABUL': tabulation of ISORROPIA (vectorized)
  CMINERAL = 'EQSAM': EQSAM parameterization (vectorized)

- CORGANIC: type of parameterization for organic gas/particle balance. To activate organic parameterization it is necessary to use a chemical scheme capable forming secondary organic aerosol (i.e. RELACS2 or CACM). Possible values are:
  CORGANIC = 'PUN' : PUN parameterization
  CORGANIC = 'MPMPO': MPMPO (non vectorized)

- CNUCLEATION: type of parameterization for nucleation (formation of new particle from sulphates). Possible values are:
  CNUCLEATION = 'KULMALA' : KULMALA parameterization
  CNUCLEATION = 'KERMINEN': KERMINEN parameterization

- Convective scavenging is activated with LCH_CONV_SCAV in NAM_CH_MNHCn.

9.2.19 Namelist NAM_CH_SOLVERn (control stiff solvers for modeln)

Fortran name	Fortran type	default value
CSOLVER	32 characters	'EXQSSA'
NSSA	integer	0
NSSAINDEX	array integers	1000*0
XRTOL	real	0.001
XATOL	real	0.1
NRELAB	integer	2
NPED	integer	1
NMAXORD	integer	5
LPETZLD	logical	TRUE
CMETHOD	1 character	N
CNORM	1 character	A
NTRACE	integer	0
XALPHA	real	0.5
XSLOW	real	100.0
XFAST	real	0.1
NQSSAITER	integer	1
XDTMIN	real	0.1
XDTMAX	real	600.
XDTFIRST	real	10.

- CSOLVER: type of numerical method used to resolve the ode system of coupling differential equations for chemistry (chemistry solver). for the description of each method, see the associated ch_routine. rosenbrock'method are gouped in mode_RBK90_Integrator routine. possible values are:
  - CSOLVER='SIS'
  - CSOLVER='LINSSA'
  - CSOLVER='CRANCK'
  - CSOLVER='QSSA'
  - CSOLVER='EXQSSA'
  - CSOLVER='ROS1'
  - CSOLVER='ROS2'
  - CSOLVER='ROS3'
  - CSOLVER='ROS4'
  - CSOLVER='RODAS3'
  - CSOLVER='RODAS4'
  - CSOLVER='ROSENBROCK': default method ROS1 with ROSENBROCK

- NSSA: number of variables to be treated as ”steady state”.
- NSSAINDEX: index set of steady state variables.
- XRTOL: relative tolerance for SVODE and D02EAF, D02EBF, D02NBF methods.
- XATOL: absolute tolerance for SVODE and D02NBF.
- NRELAB: choose relative error for NAG's D02EBF solver:
  - NRELAB=1 : for correct decimal places
  - NRELAB=2 : for correct significant digits
  - NRELAB=0 : for a mixture
- NPED: calculation parameter of the Jacobian matric for SVODE and NAG's D02EBF/D02NBF solvers:
  - NPED=1 : for analytical Jacobian (using subroutine CH_JAC)
  - NPED=0 : for numerical Jacobian
- NMAXORD: maximum order for the BDF method (0¡NMAXORD¡=5) for NAG's D02NBF solver.
- LPETZLD: switch to activate Petzold local error test (recommended) for NAG's D02NBF solver.
- CMETHOD: method to use non-linear system for NAG's D02NBF solver.
  - CMETHOD='N' or 'D' : modified Newton iteration
  - CMETHOD='F' : functional iteration
- CNORM: type of norm to be used for NAG's D02NBF solver.
  - CNORM='A' or 'D' : averaged L2 norm
  - CNORM='M' : maximum norm
- NTRACE: level of output from D02NBF solver:
  - NTRACE=-1 : no output
  - NTRACE=0 : only warnings are printed
  - NTRACE≥1 : details on Jacobian entries, nonlinear iteration and time integration are given

- XALPHA: the Cranck-Nicholson parameter (0,1).
- XSLOW: slow species, lifetime ¿ XSLOW * timestep for EXQSSA and QSSA methods.
- XFAST: fast species, lifetime ¡ XFAST * timestep for EXQSSA and QSSA methods.
- NQSSAITER: number of iterations in QSSA method.
- XDTMIN: minimal allowed timestep for EXQSSA.
- XDTMAX: maximal allowed timestep for EXQSSA.
- XDTFIRST: timestep for first integration step of EXQSSA.

9.2.20 Namelist NAM_CONDSAMP (Conditional sampling)

Fortran name	Fortran type	default value
LCONDSAMP	logical	FALSE
NCONDSAMP	integer	3
XRADIO	array(real)	3*900.
XSCAL	array(real)	3*1.
XHEIGHT_BASE	real	100.
XDEPTH_BASE	real	100.
XHEIGHT_TOP	real	100.
XDEPTH_TOP	real	100.

It contains the parameters to activate conditional sampling (Couvreux et al., 2010). The first tracer is released at the surface, the second one is released XHEIGHT_BASE below the cloud base on XDEPTH_BASE depth the third one is released XHEIGHT_TOP above the cloud top on XDEPTH_TOP depth.

- LCONDSAMP : Flag to activate conditional sampling
- NCONDSAMP : Number of conditional samplings
- XRADIO : Period of radioactive decay
- XSCAL : Scaling factor
- XHEIGHT_BASE : Height below the cloud base where the 2nd tracer is released
- XDEPTH_BASE : Depth on which the 2nd tracer is released
- XHEIGHT_TOP : Height above the cloud top where the 3rd tracer is released
- XDEPTH_TOP : Depth on which the 3rd tracer is released

9.2.21 Namelist NAM_CONF (global configuration parameters)

It contains the model configuration parameters common to all the models. They are included in the module MODD_CONF.

Fortran name	Fortran type	default value
CCONF	5 characters	'START'
LFLAT	logical	FALSE
CEQNSYS	3 characters	'DUR'
LFORCING	logical	FALSE
NMODEL	integer	1
NVERB	integer	5
NHALO	integer	1
CSPLIT	10 characters	'YSPLITTING'
LLG	logical	FALSE
LINIT_LG	logical	FALSE
CINIT_LG	5 characters	'FMOUT'
LNOMIXLG	logical	FALSE
CEXP	5 characters	'EXP01'
CSEG	5 characters	'SEG01'

- CCONF: configuration of all models
  - 'START ' for start configuration
  - 'RESTA' for restart configuration
- CEQNSYS: Equation system resolved by the MESONH model
  - 'LHE' Lipps and HEmler anelastic system
  - 'DUR' approximated form of the DURran version of the anelastic sytem
  - 'MAE' classical Modified Anelastic Equations but with not any approximation in the momentum equation
- LFLAT: Flag for zero ororography
  - .TRUE. = no orography (zs=0.)
  - .FALSE. = the orography is not zero everywhere
- LFORCING: Flag to use forcing sources
  - .TRUE. add forcing sources
  - .FALSE. no forcing sources
- NMODEL: Number of nested models

- NVERB: Level of informations on output-listing
  - 0 for minimum of prints
  - 5 for intermediate level of prints
  - 10 for maximum of prints
- NHALO: Size of the halo for parallel distribution. This variable is related to computer performance but has no impact on simulation results
- CSPLIT: Kind of domain splitting for parallel distribution. This variable is related to computer performance but has no impact on simulation results
  - 'BSPLITTING' domain is decomposed in Box along X and Y
  - 'XSPLITTING' the X direction is splitted in stripes along Y
  - 'YSPLITTING' the Y direction is splitted in stripes along X
- LLG: Flag to use Lagrangian variables
- LINIT_LG: Flag to reinitialize Lagrangian variables (with LLG=.T.)
- CINIT_LG: with LINIT_LG=T :
  - 'FMOUT' each time an output file is written
  - other string: only when starting a new segment (CCONF='RESTA')
- LNOMIXLG: Flag to unset the turbulence for LG variables.
  You must have LNOMIXLG=.TRUE. with CSCONV='EDKF'
- CEXP: Experiment name (this is the name of the set of run, you have performed or you want to perform on the same physical subject) **Please do not leave any blank character in this name!**
- CSEG: Name of segment (this is the name of the future run, you want to perform) **Please do not leave any blank character in this name!**

From these last two informations, we built the names of the different MESONH output files:

CEXP.$n.CSEG.nbr

where $n represents the number of the model which generates this output and nbr is the number of the outfile. For instance, if $CEXP =' HYDRO'$ and $CSEG =' INIT1'$ and we use only one model (no gridnesting) the different output will be called:

$HYDRO.1.INIT1.001$, $HYDRO.1.INIT1.002$, ...

9.2.22 Namelist NAM_CONFn (configuration of model n)

Fortran name	Fortran type	default value
LUSERV	logical	TRUE
LUSECI	logical	FALSE
LUSERC	logical	FALSE
LUSERR	logical	FALSE
LUSERI	logical	FALSE
LUSERS	logical	FALSE
LUSERG	logical	FALSE
LUSERH	logical	FALSE
NSV_USER	integer	0

It contains the model configuration parameters specific for the model n. They are included in the module MODD_CONFn.

- LUSERV : Flag to use vapor mixing ratio (prognostic variable r_v)
  - .TRUE. r_v is present
  - .FALSE. r_v is not allocated
- LUSECI : Flag to use Pristine Ice (diagnostic variable C_i)
  - .TRUE. C_i is present
  - .FALSE. C_i is not allocated
- LUSERC (Same as LUSERV but for the cloud mixing ratio r_c), LUSERR (for rain mixing ratio r_r), LUSERI (for ice mixing ratio r_i), LUSERS (for snow mixing ratio r_s), LUSERG (for graupel mixing ratio r_g) and LUSERH (for hail mixing ratio r_h) : **You don't need to fill this records : they are directly managed by CCLOUD.**
- NSV_USER : Number of user passive scalar variables
  Caution! Scalar variables needed for the 2-moment microphysical schemes, lagrangian trajectory, passive pollutants or the chemistry options are treated automatically by the model and should not be counted here.

9.2.23 Namelist NAM_CONFZ

See section 5.2.6 page 44 for details.

9.2.24 Namelist NAM_CONVECTn

The namelist NAM_CONVECTn does not exist any more since masdev48 : it is replaced by NAM_PARAM_KAFRn. The logicals LSHAL and LDEEP are deleted.

9.2.25 Namelist NAM_DRAGTREE

This namelist allows to take into account drag of trees in the atmospheric model instead of SURFEX according to Aumond et al. (2011) in the case of very fine vertical resolution. The Z0 vegetation is therefore reduced to the roughness of grassland in SURFEX (z0v_from_lai.F90). LTREE_DRAG in NAM_TREEDRAG of SURFEX must also be activated.

Fortran name	Fortran type	default value
LDRAGTREE	logical	FALSE

- LDRAGTREE: flag to activate drag of trees

9.2.26 Namelist NAM_DUST

This namelist is use to activate explicit aerosol dusts. It is not necessary to use chemistry to activate dusts but it is recommended to activate on-line dust emissions (see surface namelists). Radiative direct effects are automatically deduced from an interpolation table of SHDOM radiative code (Mie).

Fortran name	Fortran type	default value
LDUST	logical	FALSE
LVARSIG	logical	FALSE
LSEDIMDUST	logical	FALSE
NMODE_DST	integer	3
LRGFIX_DST	logical	FALSE
LDEPOS_DST	logical	FALSE

- LDUST: flag to activate passive dust aerosol.
- LVARSIG: flag to activate variable standard deviation for each dust mode.
- LSEDIMDUST: flag to activate dust sedimentation.
- NMODE_DST: number of lognormal dust modes (maximum of 3 modes).
- LRGFIX_DST : flag to use only 1 moment for each dust mode (LRGFIX_DST='TRUE' associated to LVARSIG='FALSE)
- LDEPOS_DST (new in masdev48) flag to activate wet dust deposition

9.2.27 Namelist NAM_DYN (global parameters for the dynamics)

Fortran name	Fortran type	default value
XSEGLEN	real	43200.
XASSELIN	real	0.2
XASSELIN_SV	real	0.02
LCORIO	logical	TRUE
LNUMDIFU	logical	FALSE
LNUMDIFTH	logical	FALSE
LNUMDIFSV	logical	FALSE
LZDIFFU	logical	FALSE
XALKTOP	real	0.01
XALZBOT	real	4000.

It contains the dynamics parameters common to all models. They are included in the module MODD_DYN.

- XSEGLEN : Segment length in seconds, corresponding to the duration of the segment simulation.
- XASSELIN : Amplitude of the Asselin temporal filter for meteorological variables
- XASSELIN_SV : Same as XASSELIN but for scalar variables
- LCORIO : Flag to set the Coriolis parameters f and f^* to zero
  - .TRUE. the Earth rotation is taken into account
  - .FALSE. the Earth rotation effects are neglected
- LNUMDIFU (formerly in masdev47 LNUMDIFF) : Flag to activate the numerical diffusion for momentum (XT4DIFU in NAM_DYNn defines the intensity of this diffusion).
- LNUMDIFTH (formerly in masdev47 LNUMDIFF) : Flag to activate the numerical diffusion for meteorological scalar variables (temperature, water substances and TKE) (XT4DIFTH in NAM_DYNn defines the intensity of this diffusion). If CMET_ADV_SCHEME is PPM_01, it is not necessary to activate numerical diffusion.
- LNUMDIFSV (formerly in masdev47 LNUMDIFF) : Same as LNUMDIFTH but for scalar variables
- LZDIFFU: Flag to apply the horizontal diffusion to potential temperature and vapor mixing ratio according to Zangl (2002) adapted to mountainous topography. No amplitude is applied for this type of diffusion.
  - .TRUE. This horizontal diffusion is applied

– .FALSE. This horizontal diffusion is not applied

This flag is independant from LNUMDIFU and LNUMDIFSV, applied to the dynamical variables and the scalar variables respectively.

- XALKTOP : Maximum value of the Rayleigh damping (in s^{-1})at the top of the upper absorbing layer. The shape of the absorbing layer is a $\sin^2$ of $\hat{z}$ (see the scientific documentation for more details).
- XALZBOT : Height (in meters) in the physical space of the upper absorbing layer base.

9.2.28 Namelist NAM_DYNn (parameters for the dynamics of model n)

Fortran name	Fortran type	default value
XTSTEP	real	60.
CPRESOPT	4 characters	'CRESI'
NITR	integer	4
LITRADJ	logical	TRUE
XRELAX	real	1.
LHORELAX_UVWTH	logical	FALSE
LHORELAX_RV	logical	FALSE
LHORELAX_RC	logical	FALSE
LHORELAX_RR	logical	FALSE
LHORELAX_RI	logical	FALSE
LHORELAX_RS	logical	FALSE
LHORELAX_RG	logical	FALSE
LHORELAX_RH	logical	FALSE
LHORELAX_TKE	logical	FALSE
LHORELAX_SV	array logical	FALSE
LHORELAX_SVC2R2	logical	FALSE
LHORELAX_SVC1R3	logical	FALSE
LHORELAX_SVLG	logical	FALSE
LHORELAX_SVCHEM	logical	FALSE
LHORELAX_SVDST	logical	FALSE
LHORELAX_SVPP	logical	FALSE
LHORELAX_SVAER	logical	FALSE
LVE_RELAX	logical	FALSE
NRIMX	integer	1
NRIMY	integer	1
XRIMKMAX	real	$1/(100 * 60.)$
XT4DIFU	real	1800.
XT4DIFTH	real	1800.
XT4DIFSV	real	1800.

It contains the specific dynamic parameters for the modesimulation.texl n. They are included in the module MODD_DYNn.

- XTSTEP : Time step in seconds. If the model is not the DAD model, XTSTEP is not taken into account but NDTRATIO in NAM_NESTING.
- CPRESOPT : Pressure solver option. 3 choices are implemented in MESONH for the moment (see the Scientific documentation for more details) :
  - 'RICHA' Richardson method preconditionned by the flat cartesian operator
  - 'CGRAD' Generalized pre-conditioned gradient for non-symmetric problems with the same preconditioner
  - 'CRESI' Conjugate Residual method

  If the problem is flat and cartesian, then the resolution becomes exact and no iteration is performed.
- NITR : Number of iterations for the pressure solver. The value of this parameter depends on the maximum slope of the orography in the model.
- LITRADJ : Logical to adjust the number of iterations for the pressure solver according to the range of the residual divergence.
- XRELAX : Relaxation coefficient in the Richardson method (CPRESOPT = 'RICHA'). This value can be less than 1 only for very steep orography, in general, the optimal value is equal to 1.
- LHORELAX_UVWTH : Flag for the horizontal relaxation applied on the outermost verticals of the model for U,V,W TH variables.
  - .TRUE. The horizontal relaxation is applied
  - .FALSE. The horizontal relaxation is not applied
- LHORELAX_RV, LHORELAX_RC, LHORELAX_RR, LHORELAX_RI, LHORELAX_RS, LHORELAX_RG, LHORELAX_RH, LHORELAX_TKE, LHORELAX_SV, LHORELAX_SVCHEM, LHORELAX_SVC2R2, LHORELAX_SVC1R3, LHOREAX_SVLG, LHORELAX_SVDST, LHORELAX_SVPP, LHORELAX_SVAER, LHORELAX_SVELEC : same as for other variables

  It is safer to set all the LHORELAX_ values rather than use their default values which can be modified by the desfm file.

- LVE_RELAX : Flag for the vertical relaxation applied to the outermost verticals of the model.
  - .TRUE. The vertical relaxation is applied
  - .FALSE. The vertical relaxation is not applied
- NRIMX : number of points in the lateral relaxation in the x axis.
- NRIMY : number of points in the lateral relaxation in the Y axis.
- XRIMKMAX : maximum value (in s^{-1}) of the relaxation coefficient for the lateral relaxation area. This value is applied to all the outermost verticals of the domain. **Caution : this value is also used to relaxe the normal wind for open lbc conditions. This relaxation exists in the Carpenter equation even if LHO_RELAX_UVWTH=F.**
- XT4DIFU (formerly in masdev47 XT4DIFF) : characteristic time (e-folding time) of the fourth order numerical diffusion for momentum (in seconds). Associated to LNUMDIFU in NAM_DYN.
- XT4DIFTH (formerly in masdev47 XT4DIFF) : characteristic time (e-folding time) of the numerical diffusion of fourth order for meteorological variables (in seconds). Associated to LNUMDIFTH in NAM_DYN.
- XT4DIFSV (formerly in masdev47 XT4DIFF) : characteristic time (e-folding time) of the numerical diffusion of fourth order for scalar variables (in seconds). Associated to LNUMDIFSV in NAM_DYN.

9.2.29 Namelist NAM_ELEC

Fortran name	Fortran type	default value
LOCG	logical	.FALSE.
LELEC_FIELD	logical	.TRUE.
LFLASH_GEOM	logical	.TRUE.
LFW_HELFA	logical	.FALSE.
LCOSMIC_APPROX	logical	.FALSE.
LION_ATTACH	logical	.TRUE.
CDRIFT	3 characters	'PPM'
LRELAX2FW_ION	logical	.FALSE.
LINDUCTIVE	logical	.FALSE.
LSAVE_COORD	logical	.FALSE.
LLNOX_EXPLICIT	logical	.FALSE.
LSERIES_ELEC	logical	.FALSE.
NTSAVE_SERIES	integer	60
NFLASH_WRITE	integer	100
CNI_CHARGING	5 characters	'TAKAH'
XQTC	real	263.
XLIM_NI_IS	real	10.E-15
XLIM_NI_IG	real	30.E-15
XLIM_NI_SG	real	100.E-15
CLSOL	5 characters	'RICHA'
NLAPITR_ELEC	integer	4
XRELAX_ELEC	real	1
XETRIG	real	200.E3
XEBALANCE	real	0.1
XEPROP	real	15.E3
XQEXCES	real	2.E-10
XQNEUT	real	1.E-10
XDFRAC_ECLAIR	real	2.3
XDFRAC_L	real	1500.
XWANG_A	real	0.34E21
XWANG_B	real	1.3E16

It contains the different parameters used by the electrical scheme. They are included in the declarative module MODD_ELEC_DESCRn.

- LOCG : when this logical swith is set to .TRUE., only the cloud electrification is computed. When set to .FALSE., lightning flashes can be produced.
- LELEC_FIELD : when this logical switch is set to .TRUE., the electric field is computed.
- LFLASH_GEOM : when this logical switch is set to .TRUE., the lightning flash branches are produced randomly. (only one lightning scheme implemented, then must be set to .TRUE.)
- LFW_HELFA : when .T. Helsdon-Farley Fair Weather field

- LCOSMIC_APPROX : .T.: Neglecting height variations of fair ion weather ion current in calculating ion source from cosmic rays
- LION_ATTACH : when .T. ion attachment to hydrometeors is considered
- CDRIFT : ion drift
  - 'PPM' : PPM advection scheme
  - 'DIV' : divergence form
- LRELAX2FW_ION : when .T. relaxation to fair weather concentration in rim zone and top absorbing layer
- LINDUCTIVE : when this logical swith is set to .TRUE., the inductive charging mechanism is taken into account.
- LSAVE_COORD : when this logical switch is set to .TRUE., the flash coordinates are written in an ascii file.
- LSERIES_ELEC : when this logical switch is set to .TRUE., some dynamical and microphysical parameters are computed and saved in an ascii file
- NTSAVE_SERIES : time interval (s) at which data from series_cloud_elec are written in an ascii file
- NFLASH_WRITE : number of flashes to be saved before writing the diag and/or coordinates in ascii files
- LLNOX_EXPLICIT : when this logical switch is set to .TRUE., nitrogen oxides are produced along the lightning path (not yet implemented)
- CNI_CHARGING : non-inductive charging parameterization
  - 'HELFA' : based on Helsdon and Farley (1987)
  - 'TAKAH' : based on Takahashi (1978)
  - 'SAUN1' : based on Saunders et al. (1991), but does not take into account the marginal positive and negative regions at low liquid water content
  - 'SAUN2' : based on Saunders et al. (1991)
  - 'SAP98' : based on Saunders and Peck (1998)
  - 'GARDI' : based on Gardiner et al. (1985)
- XQTC : temperature charge reversal (K), only if CNI_CHARGING = 'HELFA'

- XLIM_NI_IS = 2.E-15 max magnitude of dq for I-S non-inductive charging (C)
- XLIM_NI_IG = 2.E-14 max magnitude of dq for I-G non-inductive charging (C)
- XLIM_NI_SG = 5.E-14 max magnitude of dq for S-G non-inductive charging (C)
- CLSOL : Laplace equation solver for the electric field
- NLAPITR_ELEC : number of iterations for the electric field solver
- XRELAX_ELEC : relaxation factor for the electric field solver
- XETRIG : electric field threshold (V m^{-1}) for lightning flash triggering
- XEBALANCE : (1-XEBALANCE) is the proportion of XETRIG over which a lightning can be triggerred to take into account the subgrid scale variability
- XEPROP : electric field threshold (V m^{-1}) for the bidirectional leader propagation
- XQEXCES : charge density threshold (C m^{-3}) for neutralization
- XDFRAC_ECLAIR : fractal dimension of lightning flashes
- XDFRAC_L : linear coefficient for the branch number
- XWANG_A : a parameter of the Wang et al. (1998) formula for LNOx production (not yet implemented)
- XWANG_B : b parameter of the Wang et al. (1998) formula for LNOx production (not yet implemented)

9.2.30 Namelist NAM_FMOUT (output instants)

Fortran name	Fortran type	default value
XFMOUT	array (real)	8*192* 999.

- XFMOUT(m,i) is an array of increments in seconds from the beginning of the segment to the instant where the i-th fields output on FM-files is realized by model "m"

9.2.31 Namelist NAM_FRC (forcing control)

Application of a specific forcing is enabled by a dedicated flag. When a Newtonian relaxation is requested, the damping time XRELAX_TIME_FRC and the height (fixed or physically based) above which the forcing is applied, XRELAX_HEIGHT_FRC and CRELAX_HEIGHT_TYPE, must be set.

Fortran name	Fortran type	default value
LGEOST_UV_FRC	logical	FALSE
LGEOST_TH_FRC	logical	FALSE
LTEND_THRV_FRC	logical	FALSE
LVERT_MOTION_FRC	logical	FALSE
LRELAX_THRV_FRC	logical	FALSE
LRELAX_UV_FRC	logical	FALSE
XRELAX_TIME_FRC	real	10800.
XRELAX_HEIGHT_FRC	real	0.
CRELAX_HEIGHT_TYPE	character*4	'FIXE'
LTRANS	logical	FALSE
XUTRANS	real	0.
XVTRANS	real	0.

- LGEOST_UV_FRC : flag to use a prescribed geostrophic wind.
  - .TRUE. to integrate a geostrophic wind with a constant Coriolis parameter $f = 2 \times \Omega \times$SIN(XLAT0). The LCORIO flag of module MODD_DYN must be .TRUE.
  - .FALSE. not active
- LGEOST_TH_FRC : flag to apply a large scale horizontal advection on the potential temperature field. The gradients result from the thermal wind balance.
  - .TRUE. to integrate an horizontal advection of θ.
  - .FALSE. not active
- LTEND_THRV_FRC : flag to simulate a large scale θ and humidity tendency.
  - .TRUE. to integrate a tendency for θ and r_v.
  - .FALSE. not active
- LVERT_MOTION_FRC : flag to simulate a large scale vertical transport of all the prognostic fields.
  - .TRUE. to integrate a vertical transport with an upstream scheme.
  - .FALSE. not active
- LRELAX_THRV_FRC : flag to apply a Newtonian relaxation on the potential temperature and humidity fields.
  - .TRUE. to relax θ and r_v towards large scale values.
  - .FALSE. not active

- LRELAX_UV_FRC : flag to apply a Newtonian relaxation on each horizontal wind component.
  - .TRUE. to relax the horizontal wind towards large scale values.
  - .FALSE. not active
- XRELAX_TIME_FRC : constant damping time for the forced relaxation.
- XRELAX_HEIGHT_FRC : height above which a forced relaxation is enabled when CRELAX_HEIGHT_TYPE='FIXE' or minimal height if 'THGR' is used.
- CRELAX_HEIGHT_TYPE : definition of the height above which a forced relaxation is enabled.
  - 'FIXE' means that a forced relaxation is never applied below XRELAX_HEIGHT_FRC.
  - 'THGR' means that a forced relaxation is never applied below the maximal height between XRELAX_HEIGHT_FRC and the height above which $\partial\theta/\partial z$ is the highest for each column.
- LTRANS : flag to apply a Galilean translation of the domain of simulation
  - .TRUE. The translation speed of the domain of simulation will be XUTRANS,XVTRANS
  - .FALSE. : not active

9.2.32 Namelist NAM_LBCn (boundary conditions of model n)

Fortran name	Fortran type	default value
CLBCX	array(2 characters)	2*"CYCL"
CLBCY	array(2 characters)	2*"CYCL"
XCPHASE	real	20.

It contains the parameters needed to specify the lateral boundary conditions for the model n. They are included in the declarative module MODD_LBCn

- CLBCX : represent the type of lateral boundary condition at the left and right boundaries along x (CLBCX(1) and CLBCX(2) respectively). The possible values are :
  - 'CYCL' for cyclic boundary conditions (in this case CLBCX(1)=CLBCX(2)='CYCL')
  - 'OPEN' for open boundary condition (Sommerfeld equation for the normal velocity)
  - 'WALL' for wall boundary condition (zero normal velocity)

- CLBCY : array containing 2 elements: they represent the type of lateral boundary condition at the left and right boundaries along y (CLBCY(1) and CLBCY(2) respectively). The possible values are identical to those for CLBCX.

- XCPHASE : imposed phase velocity of the outgoing gravity waves. This phase velocity can be used in the Sommerfeld equation which gives the temporal evolution of the normal velosity at the open lateral boundary.

9.2.33 Namelist NAM_LES (LES budgets)

This namelist controls the diagnostics of turbulence, especially for Large Eddy Simulations. The diagnostics are saved in the diachronic file (.000). The list of the diagnostics is given in annexe C page 189.

Fortran name	Fortran type	default value
LLES_MEAN	logical	.FALSE.
LLES_RESOLVED	logical	.FALSE.
LLES_SUBGRID	logical	.FALSE.
LLES_LNOLBLBARUPDRAFT	logical	.FALSE.
LLES_DOWNDRAFT	logical	.FALSE.
LLES_SPECTRA	logical	.FALSE.
LLES_CS_MASK	logical	.FALSE.
NLES_LEVELS	integer (:)	all levels
XLES_HEIGHTS	real (:)	none
NSPECTRA_LEVELS	integer (:)	none
XSPECTRA_HEIGHTS	real (:)	none
CLES_NORM_TYPE	character (len=4)	'NONE'
CBL_HEIGHT_DEF	character (len=3)	'KE '
XLES_TEMP_SAMPLING	real	60 s if CTURB='3DIM' 300 s if CTURB='1DIM'
XLES_TEMP_MEAN_START	real	none
XLES_TEMP_MEAN_END	real	none
XLES_TEMP_MEAN_STEP	real	3600 s
LLES_CART_MASK	logical	.FALSE.
NLES_IINF	integer	physical domain boundary (JPHEXT+1)
NLES_ISUP	integer	physical domain boundary (NIMAX+JPHEXT)
NLES_JINF	integer	physical domain boundary (JPHEXT+1)
NLES_JSUP	integer	physical domain boundary (NJMAX+JPHEXT)
LLES_NEB_MASK	logical	.FALSE.
LLES_CORE_MASK	logical	.FALSE.
LLES_MY_MASK	logical	.FALSE.
NLES_MASKS_USER	integer	NUNDEF

- LLES_MEAN : flag for computation of the mean vertical profiles of the model variables

- LLES_RESOLVED : flag for computation of the mean vertical profiles of the resolved fluxes, variances and covariances
- LLES_SUBGRID : flag for computation of the mean vertical profiles of the subgrid fluxes, variances and covariances
- LLES_UPDRAFT : flag for computation of the updraft vertical profiles of some resolved and subgrid fluxes, variances and covariances
- LLES_DOWNDRAFT : Same as LLES_UPDRAFT but for downdrafts.
- LLES_SPECTRA : flag for computation of the non-local diagnostics (2 points correlations and spectra)
- LLES_CS_MASK : flag for computation of the conditional sampling diagnostics
- NLES_LEVELS : list of model levels where the local quantities are computed. Default is: all model levels (by default, the vertical profiles are computed on the MESO-NH grid).
- XLES_HEIGHTS : list of constant altitude levels where the local quantities are computed. Not used by default.
- NSPECTRA_LEVELS : list of model levels where the non-local quantities are computed. Any number is allowed, but too many will be costly in CPU time and memory.
- XSPECTRA_HEIGHTS : list of constant altitude levels where the non-local quantities are computed. Any number is allowed, but too many will be costly in CPU time and memory.
- CLES_NORM_TYPE : type of normalization for the fluxes and variances:
  - 'NONE': no normalization is computed (however, the quantities necessary to perform these are computed, and stored in the file)
  - 'CONV': convective normalization, using Q_0, w_*, h, $< \overline{w'r'_v} >_{surf}$.
  - 'EKMA': Ekman normalization, using u_* and L_{Ekman}.
  - 'MOBU': Monin-Obukhov normalization, using L_{MO}, u_*, Q_0, $< \overline{w'r'_v} >_{surf}$.
- CBL_HEIGHT_DEF : definition of the Boundary Layer height h:
  - 'KE ': test on total kinetic energy: $E(h) + e(h) = 0.05\frac{1}{h}\int_0^h (E(z) + e(z))dz$
  - 'WTV': test on $< w'\theta'_v + \overline{w'\theta'_v} >$: height h where this flux is most negative.
  - 'DTH' : test on θ profile.

- XLES_TEMP_SAMPLING : time (seconds) between two samplings of the LES profiles and non-local quantities
- XLES_TEMP_MEAN_START : time (seconds from the beginning of the simulation) at which the averaging begins. If not defined, no averaging is performed.
- XLES_TEMP_MEAN_END : time (seconds from the beginning of the simulation) at which the averaging ends. If not defined, no averaging is performed.
- XLES_TEMP_MEAN_STEP : time step (seconds) for averaging.
- LLES_CART_MASK : flag to compute the LES diagnostics only inside a cartesian subdomain defined with the indexes of the model 1. Both local and non-local quantities can be computed.
- NLES_IINF : lower i index of the cartesian subdomain. The default value is the physical domain left boundary.
- NLES_ISUP : upper i index of the cartesian subdomain. The default value is the physical domain right boundary.
- NLES_JINF : lower j index of the cartesian subdomain. The default value is the physical domain bottom boundary.
- NLES_JSUP : upper j index of the cartesian subdomain. The default value is the physical domain top boundary.
- LLES_NEB_MASK : Flag to compute the LES diagnostics separately inside and outside the model columns where clouds are present. Only local quantities can be computed.
- LLES_CORE_MASK : Flag to compute the LES diagnostics separately inside and outside the model columns where cloud core is present. Only local quantities can be computed.
- LLES_MY_MASK: Flag to compute the LES diagnostics on a mask defined by the user as a 2D horizontal mask. It must be coded at the beginning of the LES monitor routine. Only local quantities can be computed with this mask.
- NLES_MASKS_USER : number of user's masks

9.2.34 Namelist NAM_LUNITn (file names)

Fortran name	Fortran type	default value
CINIFILE	28 characters	'INIFILE'
CCPLFILE	array (28 characters)	JPCPLFILEMAX*"NONE"

It contains the names of the different files used for the initialization of the model n. They are included in the declarative module MODD_LUNITn

- CINIFILE : name of the initial FM-file which contains the field values used as initial state in the present MESONH numerical simulation
- CCPLFILE : name of the FM-files which contains the field values used for the coupling of the outermost MESONH model. No more than JPCPLFILEMAX=24 (for the present version) files can be used in a simulation. These CCPLFILE file names are only meaningful for the outermost model which finds its boundary conditions from a previously executed run of Meso-NH or another model.

  No constraint are imposed on the coupling file names only that they must be temporally ordered

  If the coupling files are given by

  CCPLFILE(1)= 'F_1' ---> t_1
  CCPLFILE(2)= 'F_2' ---> t_2
  CCPLFILE(3)= 'A_2' ---> t_3
  CCPLFILE(4)= 'A_5' ---> t_4
  CCPLFILE(5)= 'NONE' --->
  ... .
  CCPLFILE(8)= 'NONE' --->

  then, the instants must satisfy :

  $$t_{segment} \leq t_1 < t_2 < t_3 < t_4$$

  If it is not the case, the program stops. If the coupling fields are not time dependant, no coupling files are required because the coupling fields are read from the inital MESONH file of model 1 as the Larger scale fields (LSUM, LSVM, LSWM, LSTHM, LSRVM). More details can be found in the scientific documentation of the model.

9.2.35 Namelist NAM_NESTING (grid nesting configuration)

Fortran name	Fortran type	default value
NDAD	array (8 real)	m-1
NDTRATIO	array (8 integer)	1
XWAY	array (8 real)	2

- NDAD(m) : is the model number of the father of each model "m"
- NDTRATIO(m) : is the ratio between time step of model m and its father NDAD(m)
- XWAY(m) : is the interactive nesting level for model m and its father NDAD(m)
  - 1 one-way interactions

– 2 two-way interactions : upward information are given to the father (also for 2D fields (Surface precipitation and shirt wave radiative fluxes) that are used by the surface (corresponding previously to $XWAY = 3$ in masdev4_7)).

9.2.36 Namelist NAM_NUDGINGn (nudging of model n)

Fortran name	Fortran type	default value
LNUDGING	logical	.FALSE.
XTNUDGING	real	21600.

It contains the parameters needed for nudging of U,V,W,TH,Rv fields of model n towards large scale values. They are included in the declarative module MODD_NUDGINGn

- LNUDGING : flag to activate nudging for model n.
- XTNUDGING : time scale for nudging towards Large Scale values.

9.2.37 Namelist NAM_PARAMn (parameterizations' names of model n)

Fortran name	Fortran type	default value
CTURB	4 characters	'NONE'
CRAD	4 characters	'NONE'
CCLOUD	4 characters	'NONE'
CDCONV	4 characters	'NONE'
CSCONV	4 characters	'NONE'
CACTCCN	4 characters	'NONE'

It contains the different types of parameterizations used by the model n. They are included in the declarative module MODD_PARAMn.

- CTURB : type of turbulence scheme used to parameterize the transfers from unresolved scales to resolved scales.
  - CTURB = 'NONE' : no turbulence scheme.
  - CTURB = 'TKEL' : turbulence scheme with a one and a half order closure (i.e. prognostic turbulent kinetic energy (TKE) and diagnostic mixing length).
- CRAD : type of radiative transfer scheme used to parameterize the effects of the solar and infrared radiations.
  - CRAD = 'NONE' then the downward surface fluxes are set to zero
  - CRAD = 'TOPA' : the solar flux is equal to the one at TOP of Atmosphere. The infra-red flux is equal to 300 Wm^{-2}.

  - CRAD = 'FIXE' then the daily evolutions of the downward surface fluxes are prescribed. The temporal evolution is done in the routine PHYS_PARAMn by fixing the hourly value of the infrared and solar fluxes and can be modified for personnal application.
  - CRAD = 'ECMW' the ECMWF radiation scheme code is used.

- CCLOUD : type of the microphysical scheme used to parameterize the different water phases' transformations.
  - CCLOUD = 'NONE' no microphysical scheme is used. You can still use water vapor if you want (LUSERV= TRUE or FALSE)
  - CCLOUD = 'REVE' only the saturation adjustment is used to create cloud water. This liquid water is never transformed in rain water.
  - CCLOUD = 'KESS' a warm Kessler microphysical scheme is used. It allows transformations between the 3 classes of water: vapor, cloud water and rain.
  - CCLOUD = 'C2R2' a 2-moment warm microphysical scheme according to Cohard and Pinty (2000).
  - CCLOUD = 'KHKO' a 2-moment warm microphysical scheme for LES of Stratocumulus according to Khairoudinov and Kogan (2000).
  - CCLOUD = 'ICE3' a mixed microphysical scheme including ice, snow, and graupel (6 classes of hydrometeors).
  - CCLOUD = 'ICE4' same as ICE3 but with hail (7 classes of hydrometeors).

- CDCONV : type of deep convection scheme used to parameterize the effects of unresolved convective clouds.
  - CDCONV = 'NONE' : no convection scheme.
  - CDCONV = 'KAFR' : Kain-Fritsch-Bechtold scheme.

- CSCONV : type of shallow convection scheme used to parameterize the effects of unresolved shallow convective clouds.
  - CSCONV = 'NONE' : no convection scheme.
  - CSCONV = 'KAFR' : Kain-Fritsch-Bechtold scheme.
  - CSCONV = 'EDKF' : Eddy-Diffusivity-Kain-Fritsch scheme (according to Pergaud et al., 2008). Can only be used with $CTURB =' TKEL'$.

- CACTCCN : type of CCN activation scheme
  - CACTCCN = 'NONE' : no CCN activation scheme.

9.2.38 Namelist NAM_PARAM_C2R2 (control variable of the 2-moment warm microphysical schemes C2R2 and KHKO)

Fortran name	Fortran type	default value
HPARAM_CCN	character (LEN=3)	'XXX'
HINI_CCN	character (LEN=3)	'XXX'
HTYPE_CCN	character (LEN=1)	'X'
XCHEN	real	0.0
XKHEN	real	0.0
XMUHEN	real	0.0
XBETAHEN	real	0.0
XCONC_CCN	real	0.0
XR_MEAN_CCN	real	0.0
XLOGSIG_CCN	real	0.0
XFSOLUB_CCN	real	1.0
XACTEMP_CCN	real	280.0
XALPHAC	real	3.0
XNUC	real	1.0
XALPHAR	real	1.0
XNUR	real	2.0
LRAIN	boolean	TRUE
LSEDC	boolean	TRUE
LACTIT	boolean	FALSE

It contains the control parameters for the C2R2 warm microphysical scheme. They are in the declarative module MODD_PARAM_C2R2.

- HPARAM_CCN: Acronym of the CCN activation parameterization to use ('CPB', 'TFH' or 'TWO'). The 'TFH' and 'TWO' need only to prescribe the XCHEN and XKHEN parameters.
  - 'TWO' refers to the classical activation spectrum of Twomey in the form $N_{CCN}(s) = Cs^k$
  - 'TFH' includes some improvements brought by Feingold and Heymsfield (JAS, 1992) to the original activation spectrum of Twomey.
  - 'CPB' refers to an activation spectrum in the form defined in Cohard et al. (JAS, 1998) with $N_{CCN}(s) = Cs^k F(\mu, \frac{k}{2}, \frac{k}{2} + 1; -\beta s^2)$, where F is the hypergeometric function and $[C, k, \mu, \beta]$, four adjustable coefficients,
- HINI_CCN: If HPARAM_CCN='CPB' then the initial CCN characteristics are given in the 'CCN' or 'AER' format. In the 'CCN' case, the parameters XCHEN, XKHEN, XMUHEN and XBETAHEN must be given while it is the case for XCONC_CCN, XR_MEAN_CCN, XLOGSIG_CCN, XFSOLUB_CCN and XACTEMP_CCN if the 'AER' option is chosen.

  - 'CCN' The aerosols are directly characterized by their activation spectrum $N_{CCN}(s)$ in the form Cs^k or $Cs^kF(\mu, \frac{k}{2}, \frac{k}{2}+1; -\beta s^2)$.
  - 'AER' The aerosols are particles which are characterized by a lognormal distribution law in the form: $N/\sqrt{2\pi}\ln(\sigma)exp\Big(-\ln(r/\overline{r})^2/2\ln(\sigma)^2\Big)$, with distribution parameters ($\overline{r}$ is the geometric mean radius, σ the geometric standard deviation and N the total particle number), by their solubility (ϵ_m) and by their activation temperature (T) as described by Cohard et al. (JGR, 2000).

- HTYPE_CCN: Aerosol type ('M' or 'C') if HPARAM_CCN=='CPB' and HINI_CCN=='AER' is chosen.
  - 'M': NaCl composition (large size maritime aerosols)
  - 'C': $(NH_4)_2SO_4$ composition (small size continental aerosols)
- XCHEN:C parameter in the generic activation spectrum $N_{CCN}(s)$
- XKHEN: k parameter in the generic activation spectrum $N_{CCN}(s)$
- XMUHEN: μ parameter in the hypergeometric function of the CPB form of the activation spectrum $N_{CCN}(s)$
- XBETAHEN: β parameter in the hypergeometric function of the CPB form of the activation spectrum $N_{CCN}(s)$
- XCONC_CCN: aerosol concentration number (N)
- XR_MEAN_CCN: geometric mean radius of the aerosol distribution ($\overline{r}$)
- XLOGSIG_CCN: natural logarithm of the geometric standard deviation of the aerosol distribution ($\ln(\sigma)$)
- XFSOLUB_CCN: Mean solubility of the aerosols (ϵ_m)
- XACTEMP_CCN: Mean air temperature at which activation will occur.
- XALPHAC: First dispersion parameter (α_c) of the γ-distribution law of the cloud droplets ($\gamma_c(D) = \frac{\alpha_c}{\Gamma(\nu_c)}\lambda_c^{\alpha_c\nu_c}D^{\alpha_c\nu_c-1}exp(-(\lambda_cD)^{\alpha_c})$)
- XNUC: Second dispersion parameter (ν_c) of the γ-distribution law of the cloud droplets
- XALPHAR: First dispersion parameter (α_r) of the γ-distribution law of the rain drops ($\gamma_r(D) = \frac{\alpha_r}{\Gamma(\nu_r)}\lambda_r^{\alpha_r\nu_r}D^{\alpha_r\nu_r-1}exp(-(\lambda_rD)^{\alpha_r})$)
- XNUR: Second dispersion parameter (ν_r) of the γ-distribution law of the rain drops

- LRAIN: Enables the rain formation (by cloud droplet autoconversion) if set to TRUE
- LSEDC: Cloud droplets are allowed to sediment if set to TRUE
- LACTIT: Activation by radiative cooling is taken into account if set to TRUE

9.2.39 Namelist NAM_PARAM_ICE (option for the mixed phase cloud parameterization ICE3 and ICE4)

Fortran name	Fortran type	default value
LWARM	logical	.TRUE.
CPRISTINE_ICE	4 characters	'PLAT'
LSEDIC	boolean	FALSE
CSEDIM	4 characters	'SPLI'

It contains the options for the mixed phase cloud parameterizations used by the model (ICE3 or ICE4). They are included in the declarative module MODD_PARAM_ICE

- LWARM : When .TRUE. activates the formation of rain by the warm microphysical processes
- CPRISTINE_ICE : Pristine ice crystal type
  - 'PLAT' : plates
  - 'COLU' : columns
  - 'BURO' : bullet rosettes
- LSEDIC : Cloud droplets are allowed to sediment if set to TRUE
- CSEDIM (new in masdev48) : Sedimentation algorithm type
  - 'SPLI' : Splitting method (original one)
  - 'STAT' : Statistic method (accordingly to Bouteloup and Seity in AROME)

9.2.40 Namelist NAM_PARAM_KAFRn (options for the Kain-Fritsch-Bechtold convective scheme of model n)

Fortran name	Fortran type	default value
XDTCONV	real	MAX(300.0,XTSTEP)
NICE	integer	1
LREFRESH_ALL	logical	TRUE
LCHTRANS	logical	FALSE
LDOWN	logical	TRUE
LSETTADJ	logical	FALSE
XTADJD	real	3600
XTADJS	real	10800
LDIAGCONV	logical	FALSE
NENSM	integer	0

It contains the options for the Kain-Fritsch-Bechtold convection scheme (deep or shallow), used by the model n. They are included in the declarative module MODD_PARAM_KAFRn.

- XDTCONV : timestep for the call of the convective scheme. Maximum value is 300s.
- NICE : flag to include ice proceses in convection scheme (1 = yes, 0 = no ice)
- LREFRESH_ALL : flag to refresh convective columns at every call of the convection scheme.
- LCHTRANS: flag to take into account the convective transport for scalar variables (chemical variables, passive pollutants ...). Can only be used with the options CDCONV='KAFR'.
- LDOWN : flag to use downdrafts in deep convection.
- LSETTADJ : flag to allow user to define adjustment time.
- XTADJD : deep convective adjustment time (if LSETTADJ=TRUE).
- XTADJS : shallow convective adjustment time (if LSETTADJ=TRUE).
- LDIAGCONV : flag to store diagnostic variables in module MODD_DEEP_CONVECTIONn: (CAPE, deep and shallow convective cloud top and base levels, up-and downdraft mass fluxes)
- NENSM : number of additional convective ensemble members for deep convection (for the moment limited to 3)

9.2.41 Namelist NAM_PARAM_MFSHALLn (options for the Mass Flux shallow convective scheme of model n)

Fortran name	Fortran type	default value
XIMPL_MF	real	1
CMF_UPDRAFT	4 characters	'EDKF'
CMF_CLOUD	4 characters	'DIRE'
LMIXUV	Logical	TRUE
LMF_FLX	Logical	FALSE

It contains the options retained for the EDKF shallow convection scheme, used by the model n. They are included in the declarative module MODD_PARAM_MFSHALLn. Contrary to the KAFR scheme, EDKF can only be called every time step.

- XIMPL_MF : Degree of implicitness
- CMF_UPDRAFT : Type of Mass Flux Scheme ('EDKF' or 'NONE')

- CMF_CLOUD : Type of statistical cloud ('DIRE' for the direct calculation of the cloud fraction as a function of the updraft fraction or 'STAT' given by the subgrid condensation scheme)
- LMIXUV: flag to take into account the mixing on momentum
- LMF_FLX: flag to compute and store the mass fluxes on every synchronous output files

9.2.42 Namelist NAM_PARAM_RADn (options for the radiations of model n)

Fortran name	Fortran type	default value
XDTRAD	real	XTSTEP
XDTRAD_CLONLY	real	XTSTEP
CLW	4 characters	'RRTM'
CAER	4 characters	'SURF'
CEFRADL	4 characters	'MART'
CEFRADI	4 characters	'LIOU'
COPWLW	4 characters	'SMSH'
COPILW	4 characters	'EBCU'
COPWSW	4 characters	'FOUQ'
COPISW	4 characters	'EBCU'
CAOP	4 characters	'CLIM'
LCLEAR_SKY	logical	FALSE
NRAD_COLNBR	integer	1000
NRAD_DIAG	integer	0
XFUDG	real	1.

It contains the options retained for the radiations scheme, used by the model n. They are included in the declarative module MODD_PARAM_RADn.

- XDTRAD : Interval of time (in seconds) between two full radiation computations. (the radiative tendency is computed for all verticals levels). This is done to save CPU time because the radiation scheme is very expensive and the radiative tendency is not evolving too much, in some cases, during periods greater than the model timestep XTSTEP. In this case, the "radiation timestep" is increased to XDTRAD
- XDTRAD_CLONLY : Interval of time (in seconds) between two radiation computations for the cloudy columns only. This is based on the same principle as the intermittent full radiation call: the cloudy column radiative tendency may, in some cases, evolve faster than the dry ones but still slower than the timestep XTSTEP. In this case, the "cloudy radiation timestep" is increased from XDTRAD to XDTRAD_CLONLY. Of course, when all and part of the radiative tendencies must be refreshed at the same MESONH timestep, only the full radiation call is performed.

- CLW : choice of long wave radiative code
  - 'RRTM': RAPID RADIATIVE TRANSFER MODEL
  - 'MORC': MORCRETTE model
- CAER : type of aerosol distribution
  - 'SURF': deduced from cover data
  - 'TEGE': computed from Tegen et al. (1997) mensual climatology (horizontal resolution is 4 degrees of latitude by 5 degrees fo longitude
  - 'TANR': computed from ECMWF T5 climatology
  - 'NONE': no aerosol
- CEFRADL : liquid effective radius calculation
  - 'MART' : based on Martin et al. (1994, JAS)
  - 'C2R2' : based on the prediction of the number concentrations. Recommended with the 2-moment microphysical schemes.
  - 'PRES' : very old parametrization as f(pressure)
  - 'OCLN' : simple distinction between land (10) and ocean (13)
- CEFRADI : ice water effective radius calculation
  - 'LIOU' : ice particle effective radius =f(T) from Liou and Ou (1994)
  - 'SURI' : ice particle effective radius =f(T,IWC) from Sun and Rikus (1999)
  - 'C3R5' : based on the prediction of the number concentrations. Recommended with the 2-moment microphysical schemes (not yet available for mixed clouds).
  - 'FX40' : fixed 40 micron effective radius
- COPWLW : cloud water LW optical properties
  - 'SMSH': Smith-Shi formulation
  - 'SAVI': Savijarvi formulation (recommended only with 2-moment microphysical schemes with small precipitation)
  - 'MALA': Malavelle formulation (recommended only with 2-moment microphysical schemes with small precipitation)
- COPILW : ice water LW optical properties
  - 'EBCU': Ebert-Curry formulation

  - 'SMSH': Smith-Shi formulation, only with CLW='RRTM'
  - 'FULI': Fu-Liou formulation, only with CLW='MORC'

- COPWSW : cloud water short wave optical properties
  - 'FOUQ': Fouquart, 1991 formulation
  - 'SLIN': Slingo, 1989 formulation
  - 'MALA': Only for 2-moment microphysical schemes. According to Malavelle.

- COPISW : ice water short wave optical properties
  - 'EBCU': Ebert-Curry formulation
  - 'FULI': Fu-Liou formulation

- CAOP : type of aerosol optical properties calculation
  - 'CLIM': climatological aerosols
  - 'EXPL': explicit aerosols (if LORILAM=.T. in NAM_CH_ORILAM

- LCLEAR_SKY : When this flag is set to .TRUE., the radiative computations are made for a mean clear-sky and for the whole cloudy columns. This is still the way to spare some CPU time, by postulating that the clear sky columns do not lead to very different radiative tendencies. This hypothesis is only valid in academical cases.

- NRAD_COLNBR : Maximal number of air columns called by a single call of the radiative subroutine. This is performed in order to save memory, because the radiation subroutine allocate for every column of size NKMAX , NKMAX working arrays . This leads to a quadratic dependency of the memory with the number of vertical levels of the model. A way to limit the necessary memory is to split the number of columns passed to the radiation subroutine in several sets of NRAD_COLNBR column. Finally, all the desired columns (depending on the preceding parameters) will be treated by sequentially calling the radiation subroutine for every set of column.

- NRAD_DIAG : number of diagnostic fields related to the radiative scheme stored in every output synchronous files (same fields as NRAD_3D in DIAG program, p.161).

- XFUDG : subgrid cloud inhomogeneity factor.

The cloud overlap assumption is defined in the routine ini_radconf.f90. The different assumptions are :

- $NOVLP = 5$: Random overlap for Clear Sky fraction and Effective Zenithal Angle. It is the best choice without subgrid condensation.
- $NOVLP = 6$: Maximum Random Overlap for Clear Sky fraction, and Random Overlap for Effective Zenithal Angle (DEFAULT VALUE). This option is well adapted to multi-layer clouds.
- $NOVLP = 7$: Maximum overlap for Clear Sky fraction and Random Overlap for Effective Zenithal Angle. This option is well adapted in the absence of multi-layer clouds.
- $NOVLP = 8$: Maximum Random overlap for Clear Sky fraction and Effective Zenithal Angle. It corresponds to the previous configurations before masdev4_7.

9.2.43 Namelist NAM_PASPOL (Passive pollutants)

Fortran name	Fortran type	default value
LPASPOL	logical	FALSE
NRELEASE	integer	0
CPPINIT	array (3 characters)	100*'1PT'
XPPLAT	array (real)	100*0.
XPPLON	array (real)	100*0.
XPPMASS	array (real)	100*0.
XPPBOT	array (real)	100*0.
XPPTOP	array (real)	100*0.
CPPT1	array (14 characters)	100*'20010921090000'
CPPT2	array (14 characters)	100*'20010921090000'
CPPT3	array (14 characters)	100*'20010921091500'
CPPT4	array (14 characters)	100*'20010921091500'

It contains the parameters to activate passive pollutants, by specifying the position and the kinetic of the release.

- LPASPOL : Flag to activate passive pollutants
- NRELEASE : Number of releases (up to 100).
- CPPINIT : Type of initialization of the source ('1PT' or '9PT')
- XPPLAT : Latitude of the release
- XPPLON : Longitude of the release
- XPPMASS : Released mass (in g)
- XPPBOT : Height of the bottom of the release (in m)
- XPPTOP : Height of the top of the release (in m)
- CPPT1 : Starting date of the release (in YYYYMMDDHHMMSS)
- CPPT2 : Starting date of the constant rate (in YYYYMMDDHHMMSS)
- CPPT3 : Ending date of the constant rate (in YYYYMMDDHHMMSS)
- CPPT4 : Ending date of the release (in YYYYMMDDHHMMSS)

9.2.44 Namelist NAM_PDF (LES budgets)

Fortran name	Fortran type	default value
LLES_PDF	logical	.FALSE.
NPDF	integer	none
XTH_PDF_MIN	real	none
XTH_PDF_MAX	real	none
XW_PDF_MIN	real	none
XW_PDF_MAX	real	none
XTHV_PDF_MIN	real	none
XTHV_PDF_MAX	real	none
XRV_PDF_MIN	real	none
XRV_PDF_MAX	real	none
XRC_PDF_MIN	real	none
XRC_PDF_MAX	real	none
XRR_PDF_MIN	real	none
XRR_PDF_MAX	real	none
XRI_PDF_MIN	real	none
XRI_PDF_MAX	real	none
XRS_PDF_MIN	real	none
XRS_PDF_MAX	real	none
XRG_PDF_MIN	real	none
XRG_PDF_MAX	real	none
XRT_PDF_MIN	real	none
XRT_PDF_MAX	real	none
XTHL_PDF_MIN	real	none
XTHL_PDF_MAX	real	none

Each PDF includes $NPDF$ intervals number between X_PDF_MIN and X_PDF_MAX.

- LLES_PDF : Flag for pdf computation
- NPDF : Number of PDF intervals
- XTH_PDF_MIN : Minimum value of potential temperature pdf
- XTH_PDF_MAX : Maximum value of potential temperature pdf
- XW_PDF_MIN : Minimum value of vertical velocity pdf
- XW_PDF_MAX : Maximum value of vertical velocity pdf
- XTHV_PDF_MIN : Minimum value of virtual potential temperature pdf
- XTHV_PDF_MAX : Maximum value of virtual potential temperature pdf
- XRV_PDF_MIN : Minimum value of vapor mixing ratio pdf
- XRV_PDF_MAX : Maximum value of vapor mixing ratio pdf

- XRC_PDF_MIN : Minimum value of cloud mixing ratio pdf
- XRC_PDF_MAX : Maximum value of cloud mixing ratio pdf
- XRR_PDF_MIN : Minimum value of rain mixing ratio pdf
- XRR_PDF_MAX : Maximum value of rain mixing ratio pdf
- XRI_PDF_MIN : Minimum value of ice mixing ratio pdf
- XRI_PDF_MAX : Maximum value of ice mixing ratio pdf
- XRS_PDF_MIN : Minimum value of snow mixing ratio pdf
- XRS_PDF_MAX : Maximum value of snow mixing ratio pdf
- XRG_PDF_MIN : Minimum value of graupel mixing ratio pdf
- XRG_PDF_MAX : Maximum value of graupel mixing ratio pdf
- XRT_PDF_MIN : Minimum value of total mixing ratio pdf
- XRT_PDF_MAX : Maximum value of total mixing ratio pdf
- XTHL_PDF_MIN : Minimum value of θ_l pdf
- XTHL_PDF_MAX : Maximum value of θ_l pdf

9.2.45 Namelist NAM_SALT

This namelist is used to active explicit sea salt aerosols. It is not necessary to use chemistry to activate sea salt but it is recommended to activate on-line sea salt emissions (see surface namelists).

Fortran name	Fortran type	default value
LSALT	logical	FALSE
LVARSIG_SLT	logical	FALSE
LSEDIMSALT	logical	FALSE
NMODE_SLT	integer	3
LRGFIX_SLT	logical	FALSE
LDEPOS_SLT	logical	FALSE

- LSALT: flag to activate passive salt aerosol.
- LVARSIG_SLT: flag to activate variable standard deviation for each salt modes.
- LSEDIMSALT: flag to activate salt sedimentation.

- NMODE_SLT: number of lognormal salt modes (a maximum of 3 modes is allowed).
- LRGFIX_SLT: flag to use only 1 moment by salt mode (LRGFIX_SLT='TRUE' associated to LVARSIG_SLT='FALSE)
- LDEPOS_SLT: flag to activate salt wet deposition

9.2.46 Namelist NAM_TURB

It contains the characteristics of the turbulence scheme used by all models.

Fortran name	Fortran type	default value
XPHI_LIM	real	3.
XSBL_O_BL	real	0.05
XFTOP_O_FSURF	real	0.05

- 'XPHI_LIM' is the threshold value for ϕ_3 and ψ_3
- 'XSBL_O_BL' is SBL height / BL height ratio
- 'XFTOP_O_FSURF' is the fraction of surface (heat or momentum) flux used to define top of BL

9.2.47 Namelist NAM_TURB_CLOUD (mixing length for clouds)

Fortran name	Fortran type	default value
NMODEL_CLOUD	integer	999
CTURBLEN_CLOUD	4 characters	'DELT'
XCOEF_AMPL_SAT	real	5.
XCEI_MIN	real	0.001E-6
XCEI_MAX	real	0.01E-6

- NMODEL_CLOUD : model number where the modification of the mixing length in the clouds is computed,
- CTURBLEN_CLOUD : type of turbulent mixing length in the clouds ('BL89','DELT','DEAR': see CTURBLEN for meanings),
- XCOEF_AMPL_SAT : saturation of the amplification coefficient,
- XCEI_MIN : minimum threshold for the instability index (in kg/kg/m/s, beginning of the amplication),
- XCEI_MAX : maximum threshold for the instability index (in kg/kg/m/s, beginning of the saturation of the amplification).

Diagnostic quantities are written on every synchronuous files (mixing length in clear sky, mixing length modified, amplification coefficient,...) if LTURB_DIAG=.TRUE. in NAM_TURBn.

9.2.48 Namelist NAM_TURBn (turbulence parameters for model n)

Fortran name	Fortran type	default value
XIMPL	real	1.
CTURBLEN	4 characters	'BL89'
CTURBDIM	4 characters	'1DIM'
LTURB_FLX	logical	FALSE
LTURB_DIAG	logical	FALSE
LSUBG_COND	logical	FALSE
CSUBG_AUCV	4 characters	'NONE'
LSIGMAS	logical	TRUE
LSIG_CONV	logical	FALSE
LRMC01	logical	FALSE
CTOM	4 characters	'NONE'

It contains the characteristics of the turbulence scheme used by the model n. They are included in the declarative module MODD_TURBn

- XIMPL: degree of implicitness of the vertical part of the turbulence scheme. (XIMPL = 0.5 corresponds to the Cranck-Nicholson scheme for the vertical turbulent diffusion and 0. to a purely explicit scheme)
- CTURBDIM: turbulence dimensionnality.
  - CTURBDIM= '1DIM' Only the vertical turbulent fluxes are taken into account. This has to be done for relatively large horizontal gridsizes.
  - CTURBDIM= '3DIM' All the turbulent fluxes are computed, this is necessary for small horizontal gridsizes (meso-γ scales or LES)
- CTURBLEN: type of turbulent mixing length.
  - CTURBLEN='DELT' If CTURBDIM='3DIM', the cubic root of the grid volum is used in 3D simulations and the squared root of the volum in 2D simulations. If CTURBB='1DIM', we take Δz in simulation of any dimensionality. This length is always limited to $\kappa * z$ near the ground.
  - CTURBLEN='BL89' The mixing length is computed according to the Bougeault and Lacarrére scheme (refer to the scientific documentation)
  - CTURBLEN='DEAR' the mixing length is given by the mesh size depending on the model dimensionality, this length is limited to the ground distance and also by the Deardorff mixing length pertinent in the stable cases.
- LTURB_FLX: flag to compute and store all the turbulent fluxes on every output synchronous files.

- LTURB_DIAG: flag to store diagnostic quantities related to the turbulent scheme on every output synchronous files. (mesh length, Prandtl number, Schmidt number, sources of TKE...)
- LSUBG_COND: flag to activate the subgrid condensation scheme (refer to the scientific documentation for more details)
- CSUBG_AUCV (formerly LSUBG_AUCV in masdev47) : Type of subgrid autoconversion scheme.
  - 'NONE'
  - 'SIGM' for Redelsperger and Sommeria (1982) scheme using $\overline{s'r'_c}$ (if LSUBG_COND is set to TRUE and only with the mixed phase for the moment)
  - 'CLFR' from the convective cloud fraction given by EDKF (if CSCONV='EDKF' only)
- LSIGMAS: Flag for using Sigma_s from turbulence scheme instead parameterized values in ice subgrid condensation scheme
- LSIG_CONV: Flag for computing Sigma_s due to convection in ice subgrid condensation scheme
- LRMC01: Flag for computing separate mixing and dissipative length in the SBL according to Redelsperger, Mahe and Carlotti 2001
- CTOM: Consideration of Third Order Moments.
  - CTOM='NONE': No Third Order moments
  - CTOM='TM06': Parameterization of Third Order moments of heat fluxes for dry CBL according to Tomas and Masson (2006).

9.3 SURFACE SCHEMES: namelists of the externalized surface

The further definition of the surface parameters are not done by MESONH itself, but by the externalized surface included in it. Here are listed the namelists of SURFEX. See SURFEX documentation for details.

- NAM_SSO
- NAM_SURF_CSTS
- NAM_SURF_ATM

- NAM_WRITE_SURF_ATM
- NAM_SEAFLUXn
- NAM_SURF_SLT
- NAM_WATFLUXn
- NAM_FLAKEn
- NAM_TREEDRAG
- NAM_DEEPSOIL
- NAM_AGRI
- NAM_ASSIM
- NAM_SGH_ISBAn
- NAM_ISBAn
- NAM_SURF_DST
- NAM_IDEAL_FLUX
- NAM_TEBn
- NAM_CH_CONTROLn
- NAM_CH_SURFn
- NAM_CH_SEAFLUXn
- NAM_CH_WATFLUXn
- NAM_CH_TEBn
- NAM_CH_ISBAn
- NAM_CHS_ORILAM
- NAM_DIAG_SURF_ATMn
- NAM_WRITE_DIAG_SURFn
- NAM_DIAG_SURFn
- NAM_DIAG_ISBAn

- NAM_DIAG_TEBn
- NAM_DIAG_FLAKEn
- NAM_DIAG_OCEANn

9.4 Simulation on the fly of balloons or aircraft in the model fields.

In order to compare the model outputs to airborne observations and measurements, it can be interesting to simulate the deplacment of a balloon or an aircraft during the model run (in any model of gridnesting runs). A balloon is launch at a given location, and either for a particular density (iso-density balloon), a particular volume (constant volume balloon) or ascent speed (radio-sounding). In an ideal case, this suppose to be in conformal projection (LCARTESIAN=F in NAM_CONF_PRE). For iso-density balloons, initial altitude or pressure is asked. A balloon is advected by the wind of the model. It can crash. For an aircraft, the flight legs must be given by the user (location and duration).

All the prognostic fields (zonal and meridien wind (from U and V components), vertical velocity, potential temperature, pression, mixing ratios, tke, radiative surface temperature) are recorded along the trajectory of the balloon or the aircraft, as well as the trajectory itself (position in X, Y and Z directions and orography). All records are in the diachronic file (.000). Up to 30 balloons and aircraft can be used.

The specification of the flight characteristics are not given in a namelist, but directly in Fortran routines:

- *ini_balloon.f90* for balloons,
- *ini_aircraft.f90* for aircraft.

9.5 Temporal series

If you need temporal series over one or different single points, you are going to use stationn.f90/ini_stationn.f90 or profilern.f90/ini_profilern.f90 (nothing in namelist). If you need temporal series of fields averaged over a cartesian area, you are going to use seriesn.f90/ini_seriesn.f90 with LSERIES=T in NAM_SERIES and the dimensions of the area defined in NAM_SERIESn.

9.5.1 Series

Namelist NAM_SERIES

Fortran name	Fortran type	default value
LSERIES	boolean	FALSE
LMASKLANDSEA	boolean	FALSE
LWMINMAX	boolean	FALSE
LSURF	boolean	FALSE

- LSERIES : flag to write temporal series in the diachronic file (.000) of each model: evolution of horizontally and vertically averaged fields (t), evolution of horizontaly averaged vertical profiles (z,t), evolution of y-horizontaly averaged fields at one level or verticaly averaged between 2 levels (x,t).
- LMASKLANDSEA : flag to separate sea and land points in temporal series (t) and (z,t),
- LWMINMAX : flag to compute minimum and maximum of vertical velocity W in temporal serie (t).
- LSURF : flag to compute temporal series on surface fields. You have to introduce in the code the surface fields you want :
  - In get_seriesn.f90 of SURFEX : put the requested fields in ZINF. In the example of the current version XTS, XT_MNW,XT_BOT, XCT, XH_ML from modd_flaken.f90 are requested.
  - In get_surf_varn.f90 of SURFEX: adjust the tile necessary to be present (in the example PWATER is required)
  - In ini_seriesn.f90 of Meso-NH : put the number of requested fields : ex: for 5 fields, NSTEMP_SERIE1 = NSTEMP_SERIE1 +5 and give the tittle of each field
  - In seriesn.f90 of Meso-NH : give the tittle of each field

See also the namelist NAM_SERIESn.

Some examples of temporal series are available which treat pronostic fields averaged or not vertically. For other fields (for example diagnostic fields such as relative humidity), follows extrema, the following Fortran routines must be modified:

- *ini_series.f90* for initialization of size and name of diachronic records,
- *seriesn.f90* to store and possibly vertically average values during the run,
- *write_seriesn.f90* to horizontally average and write series in diachronic file.

Namelist NAM_SERIESn (temporal series in diagnostic file of model n)

Fortran name	Fortran type	default value
NIBOXL	integer	2
NIBOXH	integer	3
NJBOXL	integer	2
NJBOXH	integer	3
NKCLS	integer	2
NKCLA	integer	2
NKLOW	integer	2
NKMID	integer	2
NKUP	integer	2
NBJSLICE	integer	1
NJSLICEL	array (20 integer)	20*2
NJSLICEH	array (20 integer)	20*3
NFREQSERIES	integer	43200/(100*60)=7

- NIBOXL, NIBOXH, NJBOXL, NJBOXH: lower and upper indexes along x and y axes, respectively, of the horizontal box used to average the series (t) and (z,t).
- NKCLS, NKCLA: K level respectively in the CLS and CLA ((x,t) series of U, Rv, Rr at KCLS and W at KCLA are stored).
- NKLOW, NKUP: two K levels ((x,t) series of mean W between KLOW and KUP and mean Rc between the ground and KUP are stored).
- NKMID: a K level ((x,t) serie of Rv at KMID is stored).
- NBJSLICE: number of y-slices for (x,t) serie.
- NJSLICEL, NJSLICEH: lower and higher index along y axe for the y-slices.
- NFREQSERIES : Temporal frequency of diagnostic writing (in time step unit).

9.5.2 Profilers and stations

To compare the model outputs to observations and measurements, it can be interesting to store the simulated data for a given profiler or station. Calculations are done for all the nested models and recorded in the corresponding diachronic files. A profiler is located at a given location defined by its latitude and longitude, whereas a station is located with its latitude, longitude and altitude. Pronostic fields are recorded at a prescribed time frequency : zonal and meridien wind (from U and V components), vertical velocity, potential temperature, pression, mixing ratios, tke, radiative surface temperature. If surface diagnostics are asked (see surface namelists), surface variables (10m-wind, 2m-temperature and humidity, surface fluxes) are also recorded.

But if you want to add a new surface field, you have to introduce it in all the following routines : mnhget_surf_paramn.f90, ground_paramn.f90, ini_diag_in_run.f90, modd_diag_in_run.f90, end_diag_in_run.f90, modd_type_profiler.f90, ini_posprofilern.f90, profilern.f90 following the example to HU2M field.

The specification of the characteristics of profilers and stations are not given in a namelist, but directly in Fortran routines:

- *ini_profilern.f90* for profilers,
- *ini_stationn.f90* for stations.

9.6 Examples

FILE **EXSEG1.nam** for the previous example of prep_ideal_case

```
&NAM_LUNITn  CINIFILE = "HYD2D" /
&NAM_CONFn  LUSERV = T /
&NAM_DYNn  XTSTEP = 60., LITRADJ= T,
           LHORELAX_UVWTH = T, LVE_RELAX = T, NRIMX = 5, NRIMY = 3,
           XRIMKMAX = .00166,  XT4DIFU = 1500. /
&NAM_PARAMn  CTURB = "TKEL", CRAD = "NONE", CCLOUD = "NONE" /
&NAM_TURBn XIMPL = 1., CTURBLEN = "BL89", CTURBDIM = "1DIM", LTURB_DIAG = T,
           LTURB_FLX = T /
&NAM_LBCn CLBCX = 2*"OPEN", CLBCY = 2*"CYCL", XCPHASE = 20. /
&NAM_CONF CCONF = "START", LTHINSHELL = T, L2D = T, LFLAT = F, NMODEL = 1,
          CEQNSYS="DUR", NVERB = 1, CEXP = "EXPER", CSEG = "HYD2D" /
&NAM_DYN  XSEGLEN =20000., XASSELIN = 0.2, LCORIO = F, XALKTOP = 0.005,
          XALZBOT = 12570., LNUMDIFU =.T. /
&NAM_FMOUT XFMOUT(1,1) = 10000., XFMOUT(1,2) = 20000. /
&NAM_BLANK /
```

FILE **EXSEG1.nam** for a real case

```
&NAM_LUNITn  CINIFILE = "16J36.1.12B18.001",
              CCPLFILE(1) = "16JAN_06_MNH" /
&NAM_CONFn  LUSERV = T, LUSERC = T, LUSERR = T ,
             LUSERI = T, LUSERS = T, LUSERG =T, LUSECI= T /
&NAM_DYNn  XTSTEP = 75.,  CPRESOPT = "RICHA", NITR = 8,
            LHORELAX_UVWTH = T, LHORELAX_RV = T, LVE_RELAX = T,
            NRIMX = 5, NRIMY = 5, XRIMKMAX = 0.0083, XT4DIFU = 5000. /
&NAM_ADVn  CMET_ADV_SCHEME = "PPM_00", CSV_ADV_SCHEME = "PPM_00" /
&NAM_PARAMn  CCLOUD = "ICE3",CTURB = "TKEL", CRAD = "ECMWF"
              CGROUND = "ISBA", CDCONV = "KAFR", CSCONV="EDKF" /
&NAM_PARAM_RADn  XDTRAD = 300., XDTRAD_CLONLY = 150.,
                  NRAD_COLNBR = 400 CAER='TANR' CLW="MORC"/
&NAM_PARAM_KAFRn XDTCONV = 300., NICE = 1 LREFRESH_ALL = T, LDOWN = T /
&NAM_PARAM_GROUNDN CROUGH='Z01D' /
&NAM_LBCn  CLBCX = 2*"OPEN", CLBCY = 2*"OPEN" /
&NAM_TURBn  CTURBLEN = "BL89", CTURBDIM = "1DIM", LSUBG_COND=.T., CSUBG_AUCV="CLFR",
           LTURB_DIAG=.FALSE.,  LTURB_FLX=.FALSE. LSIG_CONV=F, LSIGMAS=T /
&NAM_CONF  CCONF = "RESTA", NVERB=2,
            NMODEL = 2, CEXP = "16J36", CSEG = "12B18" /
&NAM_DYN  XSEGLEN = 800., LCORIO = T, LNUMDIFU = T,XALKTOP = 0.001, XALZBOT = 14500. /
&NAM_NESTING NDAD(2) = 1, NDTRATIO(2) = 3, XWAY(2) = 3. /
&NAM_FMOUT XFMOUT(1,1) = 10800., XFMOUT(1,2) = 21600. ,
           XFMOUT(2,1) = 10800., XFMOUT(2,2) = 21600. /

&NAM_ISBAn CRUNOFF = "WSAT", CSCOND  = "NP89", CALBEDO = "DRY",
       CC1DRY     = 'DEF', CSOILFRZ   = 'DEF', CDIFSFCOND = 'DEF',
       CSNOWRES   = 'DEF', CROUGH = 'Z04D'/
&NAM_SEAFLUXn CSEA_ALB="UNIF", CSEA_FLUX="DIRECT"/
&NAM_DIAG_SURFn /
&NAM_DIAG_ISBAn /
&NAM_DIAG_SURF_ATMn /
```

The five latest namelists are for the externalised surface.

Chapter 10

Compute diagnostics after a MESO-NH simulation

10.1 Presentation

After running the model, useful quantities can be diagnosed from prognostic variables contained in the FM output files. It is done by the program **DIAG** which computes diagnostic variables.

Available diagnostics are listed in section 10.2.

10.1.1 The namelist file DIAG1.nam

The `DIAG1.nam` namelist file contains the diagnostics required by the user, the name of the input FM files, the suffix of the output diachronic files, and output file type. The user can reset options for the convective and radiation scheme with NAM_PARAM_KAFRn and NAM_PARAM_RADn namelist (see chapter 9).

1. Namelist NAM_DIAG (controls diagnostic variables)

   see section 10.2 for the list of all the keywords.

2. Namelist NAM_DIAG_BLANK

Fortran name	Fortran type	default value
XDUMMY_DIAG	array(real)	20* 0.
NDUMMY_DIAG	array(integer)	20* 0
LDUMMY_DIAG	array(logical)	20* TRUE
CDUMMY_DIAG	array(80 characters)	20* ”

   similar use than NAM_BLANK (see section 5.2.2 page 41). Add USE MODD_DIAG_BLANK in a diag subroutine to use any of these variables.

3. Namelist NAM_DIAG_FILE

Fortran name	Fortran type	default value
YINIFILE	array of character (len=28)	24*' '
YSUFFIX	character (len=5)	'_DIAG'

- YINIFILE : name of the input FM files.
- YSUFFIX : suffix appended to input file name to form output file name.

4. Namelist NAM_STO_FILE (controls trajectories computation)
   only read if LTRAJ=.TRUE. in NAM_DIAG

Fortran name	Fortran type	default value
CFILES	array of character (len=28)	100*' '
NSTART_SUPP	array of integer	100*NUNDEF

- CFILES : name of all the input FM files used to compute trajectories. They must be in **inverse** chronological order, and correspond to a reinitialisation of Lagrangian tracers (see chapter 9).
- NSTART_SUPP : extra origins for trajectory computations. In the second example of 10.4, the output files will contain the set of variables (X000, Y000, Z000, TH000, RV000) with origin corresponding to the last file (CFILES(6)), and 2 extra sets (X`n`, Y`n`, Z`n`, TH`n`, RV`n`) with `n`=001 for origin corresponding to the CFILES(4), `n`=002 corresponding to CFILES(2). (Note that extra origins are in chronological order).

5. Namelist NAM_PARAM_KAFRn (options for the convective scheme when convective diagnostics with NCONV_KF)
   see chapter 9 for variable meanings.

6. Namelist NAM_PARAM_RADn (options for the radiative budget when radiation diagnostics with NRAD_3D)
   see chapter 9 for variable meaning.

7. Namelists of the externalized surface
   See section 10.3 for details.

10.2 Variables available in the output diachronic file

10.2.1 Variables by default

ZS : [2D] orography (m)
ZSMT : [2D] smoothed orography for SLEVE vertical coordinate (m)
RHODREF : [3D] Dry density for reference state with orography (kg/m^3)
THVREF : [3D] Thetav for reference state with orography (K)
RHOREFZ : [1D] rhodz for reference state without orography (kg/m3)
THVREFZ : [1D] thetavz for reference state without orography (K)
EXNTOP : Exner function at model top
PPTn : [3D] passive pollutant n concentration (g/m^3) only if LPASPOL=T in YINIFILE.des

Diagnostic relative to surface

Only available if CSURF='EXTE' in YINIFILE.des

UM10 VM10 : [2D] components of wind at 10m (m/s)
FF10MAX : [2D] Wind gusts at 10 m (only if CTURB='TKEL')
SFCO2 : [2D] CO2 flux (mg/m2/s) (if present in YINIFILE)
SW : [2D] SW (W/m2)
LW : [2D] LW (W/m2)

10.2.2 General variables

Fortran name in &NAM_DIAG	Possible Values	Variables [dim] meaning (unit)
CISO	'PR'	PABSM : [3D] pression (Pa)
	'TK'	THM : [3D] potential temperature (K)
	'EV'	POVOM : [3D] potential vorticity (PVU)
	'PRTK'	PABSM + THM
	'PREV'	PABSM + POVOM
	'TKEV'	THM + POVOM
	'PREVTK'	PABSM + POVOM + THM (by default)
LVAR_RS	.FALSE.	no field
	.TRUE.	UM, VM, WM : [3D] wind components (m/s)
		RVM : [3D] water vapor mixing ratio (kg/kg)
		with LWIND_ZM=.TRUE. UM_ZM VM_ZM [3D] :Zonal and meridian components of horizontal wind (M/S)

LVAR_LS	**.FALSE.**	by default : no field
	.TRUE.	LSUM, LSVM, LSWM LSTHM,LSRVM : [3D] large scale variables
		with LWIND_ZM=.TRUE. : LSUM_ZM LSVM_ZM [3D] : Large scale zonal and meridian components of horizontal wind
LVAR_FRC (if LFORCING=T in YINIFILE.des)	**.FALSE.**	by default : no field
	.TRUE.	UFRCn : [1D] zonal component of horizontal forcing wind (m/s)
		VFRCn : [1D] meridian component of horizontal forcing wind (m/s)
		WFRCn : [1D] vertical forcing wind (m/s)
		THFRCn : [1D] θ_{frc} forcing potential temperature (K)
		RVFRCn : [1D] $(\partial r_v/\partial t)_{frc}$ forcing vapor mixing ratio (kg/kg)
		TENDTHFRCn : [1D] $(\partial \theta/\partial t)_{frc}$ (K/s)
		TENDRVFRCn : [1D] $(\partial r_v/\partial t)_{frc}$ ((kg/kg)/s)
		GXTHFRCn: [1D] $(\partial \theta/\partial x)_{frc}$ (K/m)
		GYRVFRCn: [1D] $(\partial \theta/\partial y)_{frc}$ (K/m)
		PGROUNDFRCn: [0D] forcing ground pressure (Pa)
LTPZH	**.FALSE.**	by default : no field
	.TRUE.	TEMP: [3D] Temperature (C)
		PRES : [3D] Pressure (hPa)
		ALT : [3D] height of model levels (geopotentiel in pressure level) (m)
		REHU : [3D] Relative Humidity (%) (if LUSERV=T)
		VPRES : [3D] Vapor Pressure (hPa) (if LUSERV=T)
LCOREF	**.FALSE.**	by default : no field
	.TRUE.	COREF : [3D] Refraction coindex (if LUSERV=T)
		MCOREF : [3D] modified refraction coindex (if LUSERV=T)
LMOIST_V	**.FALSE.**	by default : no field
	.TRUE.	THETAV : [3D] Virtual potential Temperature (K)
		POVOV : [3D] Virtual Potential Vorticity (PVU)
		with **LMEAN_POVO=T** MEAN_POVOV : [2D] Mean Virtual Potential Vorticity (PVU)
LMOIST_E	**.FALSE.**	by default : no field
	.TRUE.	THETAE : [3D] Equivalent potential Temperature (K)
		POVOE : [3D] Equivalent Potential Vorticity (PVU)
		with **LMEAN_POVO=T** MEAN_POVOE : [2D] Mean Equivalent Potential Vorticity (PVU)
LMEAN_POVO	**.FALSE.**	by default : no field
	.TRUE.	MEAN_POVO : [2D] Mean Potential Vorticity (PVU)
		with **LMOIST_V=T** MEAN_POVOV : [2D] Mean Virtual Potential Vorticity (PVU)
		with **LMOIST_E=T** MEAN_POVOE : [2D] Mean Equivalent Potential Vorticity (PVU)
XMEAN_POVO (1:2)	(15000,50000)	averaged between two isobaric levels in Pa (XMEAN_POVO(1),XMEAN_POVO(2))

LVORT	**.FALSE.**	by default : no field
	.TRUE.	ABVOR : [3D] vertical component of Absolute Vorticity (/s)
		UM1, VM1, WM1 : [3D] relative vorticity components (/s)
		with **LWIND_ZM=T** UM1_ZM, VM1_ZM : [3D] Zonal and Meridian components of horizontal vorticity (M/S) (/s)
LDIV	**.FALSE.**	by default : no field
	.TRUE.	HDIV : [3D] Horizontal divergence (/s)
		HMDIV : [3D] Horizontal Moisture divergence (kg/m^3/s)
LGEO	**.FALSE.**	by default : no field
	.TRUE.	UM88, VM88, WM88 : [3D] Geostrophic wind components (m/s)
		with **LWIND_ZM=T** UM88_ZM, VM88_ZM : [3D] Zonal and Meridian components of Geostrophic wind (m/s)
LAGEO	**.FALSE.**	by default : no field
	.TRUE.	UM89, VM89, WM89 : [3D] Ageostrophic wind components (m/s)
		with **LWIND_ZM=T** UM89_ZM, VM89_ZM : [3D] Zonal and Meridian components of Ageostrophic wind (m/s)
LMSLP	**.FALSE.**	by default : no field
	.TRUE.	MSLP : [2D] Mean Sea Level Pressure (hPa)
LTHW	**.FALSE.**	by default : no field
	.TRUE.	THVW : [2D] Thickness of Vapor Water (mm) (if LUSERV=T)
		THCW : [2D] Thickness of Cloud Water (mm) (if LUSERC=T)
		THRW : [2D] Thickness of Rain Water (mm) (if LUSERR=T)
		THIC : [2D] Thickness of Ice (mm) (if LUSERI=T)
		THSN : [2D] Thickness of Snow (mm) (if LUSERS=T)
		THGR : [2D] Thickness of Graupel (mm) (if LUSERG=T)
		THHA : [2D] Thickness of Hail (mm) (if LUSERH=T)
LBV_FR	**.FALSE.**	by default : no field
	.TRUE.	BV : [3D] Brunt-Vaissala frequency (/s)
		BVE : [3D] Equivalent Brunt-Vaissala frequency (/s)
LVAR_MRW	**.FALSE.**	by default : no field
	.TRUE.	MRV : [3D] Mixing Ratio for Vapor (g/kg)(if LUSERV=T)
		MRC : [3D] Mixing Ratio for Cloud (g/kg) (if LUSERC=T)
		MRR : [3D] Mixing Ratio for Rain (g/kg) (if LUSERR=T)
		MRI : [3D] Mixing Ratio for Ice (g/kg) (if LUSERI=T)
		CIT : [3D] Ice concentration (m−3 (if LUSECI=T)
		MRS : [3D] Mixing Ratio for Snow (g/kg) (if LUSERS=T)
		MRG : [3D] Mixing Ratio for Graupel (g/kg) (if LUSERG=T)
		MRH : [3D] Mixing Ratio for Hail (g/kg) (if LUSERH=T)
		CCCNM : [3D] if CCLOUD='C2R2'
		CCLOUDM : [3D] if CCLOUD='C2R2'
		RAINM : [3D] if CCLOUD='C2R2'
		CICEM : [3D] if CCLOUD='C1R3'
		CINM : [3D] if CCLOUD='C1R3'
LVAR_MRSV	**.FALSE.**	by default : no field
	.TRUE.	MRSVnnn : [3D] Mixing Ratio for User Scalar Variable n (g/kg)

LBLTOP	**.FALSE.**	by default : no field
	.TRUE.	`HBLTOP`: [2D] Height of boundary layer top (m)
		`KBLTOP`: [2D] Index of boundary layer top
		`FREE_ATM_GR`: [2D] Gradient of free atmosphere above BL top (K/m)
		`THV_FREE`: [3D] Thetav above BL top (K)
LVAR_PR	**.FALSE.**	by default : no field
	.TRUE.	`ACPRR` : [2D] (if LUSERR=T) Accumulated from the beginning of the simulation explicit Precipitation Rates (mm)
		`INPRR` : [2D] (if LUSERR=T) Instantaneous explicit Precipitation Rate (mm/h)
		`INPRR3D` : [3D] (if LUSERR=T) Instantaneous explicit 3D Rain Precipitation flux (m/s)
		`EVAP3D` : [3D] (if LUSERR=T) Instantaneous 3D Rain Evaporation flux (kg/kg/s)
		`ACPRC` : [2D] (if LUSERC=T) Accumulated Cloud Precipitation Rate (mm)
		`INPRC` : [2D] (if LUSERC=T) Instantaneous Cloud Precipitation Rate (mm/h)
		`ACPRS` : [2D] (if LUSERS=T) Accumulated explicit Precipitation Rate for Snow (mm)
		`INPRS` : [2D] (if LUSERS=T) Instantaneous explicit Precipitation Rate for Snow (mm/h)
		`ACPRG` : [2D] (if LUSERG=T) Accumulated explicit Precipitation Rate for Graupel (mm)
		`INPRG` : [2D] (if LUSERG=T) Instantaneous explicit Precipitation Rate for Graupel (mm/h)
		`ACPRH` : [2D] (if LUSERH=T) Accumulated explicit Precipitation Rate for Hail (mm)
		`ACPRH, INPRH` : [2D](if LUSERH=T) Instantaneous explicit Precipitation Rate for Hail (mm/h)
		`ACPRT` :(if LUSERR=T) [2D] Total Accumulated explicit Precipitation Rate (mm)
		`INPRT` : [2D] (if CCLOUD≠'NONE') Total Instantaneous explicit Precipitation Rate (mm/h)
		`PACCONV` : [2D] (if CDCONV≠'NONE') Convective Accumulated Precipitation Rate (mm)
		`PRCONV` : [2D] (if CDCONV≠'NONE') Convective Instantaneous Precipitation Rate (mm/h)
		`PRSCONV` : [2D] (if CDCONV≠'NONE') (mm/h) Convective instantaneous Precipitation Rate for Snow
LTOTAL_PR	**.FALSE.**	by default : no field
	.TRUE.	`ACTOPR` : [2D] Accumulated Total explicit Precipitation (mm)
		`INTOPR` : [2D] Instantaneous Total explicit Precipitation (mm/h)
		with **LMEAN_PR=T** `LS_ACTOPR` `LS_INTOPR` } [2D] Total Precipitations averaged in a Large Scale grid mesh
XMEAN_PR	(1,1)	nb of grid points of the small-scale model inside the LS grid mesh along x, y for LMEAN_PR

LCLD_COV (if LUSERC=T)	**.FALSE.**	by default : no field
	.TRUE.	`HECL` : [2D] Height of Explicit CLoud top (km)
		`HCL` : [2D] Height of maximum CLoud top (km)
		`TCL` : [2D] Temperature of maximum Cloud top
		`CLDFR` : [3D] Cloud Fraction (_)
		`VISI_HOR` : [3D] Visibility (m)

NCAPE (if LUSERV=T)	**-1**	by default : no field
	0	`CAPEMAX` : [2D] maximum of `CAPE3D` (J/kg)
		`CINMAX` : [2D] value of `CIN3D` corresponding to `CAPEMAX` (J/kg)
	1	`CAPEMAX CINMAX`
		`CAPE3D` : [3D] Convective Available Potential Energy (J/kg)
		`CIN3D` : [3D] Convective INhibition energy (J/kg)
		`DCAPE3D` : [3D] Downdraft cape (J/kg)
	2	`CAPEMAX CINMAX CAPE3D, CIN3D, DCAPE3D`
		`VKE` : [3D] Vertical Kinetic Energy (from explicit vertical motion) (J/kg)

10.2.3 Convective scheme KAFR

NCONV_KF	**-1**	default value : no fields
	0	`CAPE` : [2D] Convective Available Potentiel Energy (J/kg)
		`CLTOPCONV` : [2D] top of convective clouds(km)
		`CLBASCONV` : [2D] base of convective clouds(km)
	1	`CAPE CLTOPCONV CLBASCONV`
		`DTHCONV` : [3D] Convective tendency for potential temperature (K/s)
		`DRVCONV`: Convective tendency for vapor (/s)
		`DRCCONV`: Convective tendency for cloud (/s)
		`DRICONV` : Convective tendency for ice (/s)
		`DSVCONVnnn` :Convective tendency for scalar variable `n` (/s)
	2	`CAPE CLTOPCONV CLBASCONV DTHCONV DRVCONV DRCCONV DRICONV DSVCONVnnn`
		`UMFCONV` : [3D] Updraft Convective Mass Flux (m^2 kg/s)
		`DMFCONV` : [3D] Downdraft Convective Mass Flux (m^2 kg/s)
		`PRLFLXCONV`: [3D] Liquid PRecipitation Convective FLuX (m/s)
		`PRSFLXCONV` : [3D] Solid PRecipitation Convective FLuX (m/s)

10.2.4 Mass Flux Shallow Convection scheme

LMFFLX	**.FALSE.**	by default : no field
	.TRUE.	MF_THW_FLX : [3D] conservative potential temperature vertical flux (K*m/s)
		MF_RCONSW_FLX : [3D] conservative mixing ratio vertical flux (kg/kg*m/s)
		MF_THVW_FLX : [3D] theta_v vertical flux (K*m/s)
		MF_UW_VFLX : [3D] U momentum vertical flux (m^2/s^2)
		MF_VW_VFLX : [3D] V momentum vertical flux (m^2/s^2)

10.2.5 Turbulent scheme

LVAR_TURB	**.FALSE.**	by default : no field
	.TRUE.	TKEM : [3D] Turbulent Kinetic Energy (m^2/s^2)
		SIGS : [3D] Sigma_s from turbulence scheme (kg/kg^2)
		SRCM : [3D] Normalized 2nd_order moment (kg/kg^2)
		BL_DEPTH : [3D] Boundary Layer Depth if CTOM='TM06' (m)

LTURBDIAG	**.FALSE.**	by default : no field
	.TRUE.	AMOIST : [3D] (m) See Scientific documentation part III, chap 7 equation 7.29
		ATHETA : [3D] (m) See Scientific documentation part III, chap 7 equation 7.30
		RED_TH1, RED_R1, RED2_TH3, RED2_R3, RED2_THR3 : [3D] Redelsperger numbers
		DP: [3D] dynamical production of TKE (m^2/s^3)
		TP: [3D] thermal production of TKE (m^2/s^3)
		TR: [3D] transport of TKE (m^2/s^3)
		DISS: [3D] dissipation of TKE (m^2/s^3)
		LM_CLEAR_SKY : [3D] mixing length in clear sky (m)
		COEF_AMPL : [3D] amplification of the mixing length (_)
		LM_CLOUD : [3D] mixing length in the clouds (m)
		LM : [3D] mixing length (m)
		THLM :[3D] conservative potential temperature (K)
		RNPM :[3D] conservative mixing ratio (kg/kg)
		RVCI : [3D] $rv + rc + ri$ (kg/kg)
		GX_RVCI,GY_RVCI : [3D] x and y gradient of RVCI (kg/kg/m)
		GNORM_RVCI : [3D] Horizontal norm of the gradient of RVCI (kg/kg/m)
		QX_RVCI : [3D] x gradient of the advection of RVCI (kg/kg/m)
		QY_RVCI : [3D] y gradient of the advection of RVCI (kg/kg/m)
		QNORM_RVCI : [3D] Horizontal norm of the gradient of the advection of RVCI (kg/kg/m)
		CEI : [3D] Cloud entrainment instability index (kg/kg/m/s)

LTURBFLX	**.FALSE.**	by default : no field	
	.TRUE.	PHI3 : [3D] Turbulent Prandtl number (_)	
		PSI3 : [3D] Turbulent Schmidt number (_)	
		PSI_SV_n : [3D] Turbulent Schmidt number for the scalar variables (_)	
		THW_FLX: [3D] theta vertical flux (K*m/s) RCONSW_FLX : [3D] rv vertical flux (kg*m/s/kg) RCW_FLX : [3D] liquid water mixing ratio vertical flux (kg*m/s/kg) THL_VVAR: [3D] $< THl, THl >$ (K^2) THLRCONS_VCOR : [3D] $< THl, Rnp >$(K*kg/kg) RTOT_VVAR : [3D] $< Rnp, Rnp >$ ($(kg/kg)^2$) UW_VFLX, VW_VFLX : [3D] wind component vertical flux (m^2/s^2) WSV_FLX_n : [3D] $< W, SVth >$(SVunit m/s)	1D scheme turbulent fluxes
		U_VAR : [3D] U variance ($(m/s)^2$) V_VAR : [3D] V variance ($(m/s)^2$) W_VAR : [3D] W variance ($(m/s)^2$) UV_FLX: [3D] $< U, V >$($(m/s)^2$,) UW_HFLX : [3D] $< U, W >$ ($(m/s)^2$) VW_HFLX : [3D] $< V, W >$ ($(m/s)^2$) USV_FLX_n : [3D] $< U, SVth >$ (SVunit m/s) VSV_FLX_n : [3D] $< V, SVth >$ (SVunit m/s) THL_HVAR : [3D] $< THl, THl >$ (K^2) THLR_HCOR : [3D] $< THl, Rnp >$ (K*kg/kg) R_HVAR : [3D] $< Rnp, Rnp >$ ($(kg/kg)^2$) UTHL_FLX : [3D] horizontal $< U, THl >$ (K*m/s) VTHL_FLX: [3D] horizontal $< V, THl >$ (K*m/s) UR_FLX : [3D] horizontal $< U, Rnp >$ (kg/kg*m/s) VR_FLX : [3D] horizontal $< V, Rnp >$ (kg/kg*m/s)	3D scheme turbulent fluxes

10.2.6 Radiation scheme

NRAD_3D (Only available if CRAD≠'NONE')	**-1**	default value : no field
	0	`DTHRAD` : [3D] Radiative heating/cooling rate (K/s)
		`FLALWD` : [2D] Downward LW on FLAT surface (W/m^2)
		`DIRFLASWD` : [2D] Direct Downward SW on FLAT surface (W/m^2)
		`SCAFLASWD` : [2D] Scattered Downward SW on FLAT surface (W/m^2)
		`DIRSRFSWD` : [2D] Direct Downward SW (W/m^2)
		`CLEARCOL_TM1` : [2D] trace of cloud (_)
		`EMIS` : [2D] emmissivity (_)
		`ZENITH` : [2D] solar zenithal angle (RAD)
		`AZIM`: [2D] azimuthal angle (RAD)
		`DIR_ALB`: [2D] direct albedo(_)
		`SCA_ALB`: [2D] scattered albedo (_)
		`TSRAD`: [2D] radiative surface temperature (K)
	1	`DTHRAD FLALWD DIRFLASWD SCAFLASWD DIRSRFSWD CLEARCOL_TM1 EMIS ZENITH AZIM DIR_ALB SCA_ALB TSRAD`
		`SWF_DOWN`: [3D] Downward SW radiative fluxes (W/m^2)
		`SWF_UP`: [3D] Upward SW radiative fluxes (W/m^2)
		`LWF_DOWN`: [3D] Downward LW radiative fluxes (W/m^2)
		`LWF_UP` : [3D] Upward LW radiative fluxes (W/m^2)
		`LWF_NET` : [3D] Total SW net radiative fluxes (W/m^2)
		`SWF_NET` : [3D] Total LW radiative fluxes (W/m^2)
		`DTRAD_LW` : [3D] LW radiative tendency for `T` (K/day)
		`DTRAD_SW` : [3D] SW radiative tendency for `T` (K/day)
		`RADSWD_VIS` : [2D] surface radiative flux in visible (W/m^2)
		`RADSWD_NIR` : [2D] surface radiative flux in near-infrared (W/m^2)
		`RADLWD` : [2D] LW surface radiative flux (W/m^2)
	2	`DTHRAD FLALWD DIRFLASWD SCAFLASWD DIRSRFSWD CLEARCOL_TM1 EMIS ZENITH AZIM DIR_ALB SCA_ALB TSRAD SWF_DOWN SWF_UP LWF_DOWN LWF_UP LWF_NET SWF_NET DTRAD_LW DTRAD_SW RADSWD_NIR RADLWD`
		Clear Sky results : `SWF_DOWN_CS SWF_UP_CS LWF_DOWN_CS LWF_UP_CS LWF_NET_CS SWF_NET_CS DTRAD_LW_CS DTRAD_SW_CS RADSWD_NIR_CS RADLWD_CS`
	3	`DTHRAD FLALWD DIRFLASWD SCAFLASWD DIRSRFSWD CLEARCOL_TM1 EMIS ZENITH AZIM DIR_ALB SCA_ALB TSRAD SWF_DOWN SWF_UP LWF_DOWN LWF_UP LWF_NET SWF_NET DTRAD_LW DTRAD_SW RADSWD_NIR RADLWD SWF_DOWN_CS SWF_UP_CS LWF_DOWN_CS LWF_UP_CS LWF_NET_CS SWF_NET_CS DTRAD_LW_CS DTRAD_SW_CS RADSWD_NIR_CS RADLWD_CS`
		`PLAN_ALB_VIS` : [2D] planetary albedo in visible (_)
		`PLAN_ALB_NIR` : [2D] planetary albedo in near-infrared (_)
		`PLAN_TRA_VIS` : [2D] planetary transmission in visible(_)
		`PLAN_TRA_NIR` : [2D] planetary transmission in near-infrared (_)
		`PLAN_ABS_VIS` : [2D] planetary absorption in visible (_)
		`PLAN_ABS_NIR` : [2D] planetary absorption in near-infrared (_)

NRAD_3D	4	DTHRAD FLALWD DIRFLASWD SCAFLASWD DIRSRFSWD CLEARCOL_TM1 EMIS ZENITH AZIM DIR_ALB SCA_ALB TSRAD SWF_DOWN SWF_UP LWF_DOWN LWF_UP LWF_NET SWF_NET DTRAD_LW DTRAD_SW RADSWD_NIR RADLWD SWF_DOWN_CS SWF_UP_CS LWF_DOWN_CS LWF_UP_CS LWF_NET_CS SWF_NET_CS DTRAD_LW_CS DTRAD_SW_CS RADSWD_NIR_CS RADLWD_CS PLAN_ALB_VIS PLAN_ALB_NIR PLAN_TRA_VIS PLAN_TRA_NIR PLAN_ABS_VIS PLAN_ABS_NIR
		EFNEB_UP : [3D] upward equivalent emissivity (Morcrette scheme)(_)
		EFNEB_DOWN : [3D] downward equivalent emissivity (_)
		FLWP : [3D] liquid water path (g/m^2)
		FIWP : [3D] ice water path (g/m^2)
		EFRADL : [3D] cloud liquid water effective radius (μm)
		EFRADI : [3D] cloud ice effective radius (μm)
		SW_NEB RRTM_LW_NEB : [3D] effective cloud fraction (_)
		OTH_VIS OTH_NI1 OTH_NI2 OTH_NI3 : [3D] cloud optical thickness (_)
		SSA_VIS SSA_NI1 SSA_NI2 SSA_NI3: [3D] cloud single scattering albedo (_)
		ASF_VIS ASF_NIR1 ASF_NIR2 ASF_NIR3 : [3D] cloud asymetry factor (_)
		ODAER_VIS ODAER_NIR1 ODAER_NIR2 ODAER_NIR3 :[3D]
		SSAAER_VIS SSAAER_NIR1 SSAAER_NIR2 SSAAER_NIR3 :[3D]
		GAER_VIS GAER_NIR1 GAER_NIR2 GAER_NIR3 :[3D]
	5	DTHRAD FLALWD DIRFLASWD SCAFLASWD DIRSRFSWD CLEARCOL_TM1 EMIS ZENITH AZIM DIR_ALB SCA_ALB TSRAD SWF_DOWN SWF_UP LWF_DOWN LWF_UP LWF_NET SWF_NET DTRAD_LW DTRAD_SW RADSWD_NIR RADLWD SWF_DOWN_CS SWF_UP_CS LWF_DOWN_CS LWF_UP_CS LWF_NET_CS SWF_NET_CS DTRAD_LW_CS DTRAD_SW_CS RADSWD_NIR_CS RADLWD_CS PLAN_ALB_VIS PLAN_ALB_NIR PLAN_TRA_VIS PLAN_TRA_NIR PLAN_ABS_VIS PLAN_ABS_NIR EFNEB_DOWN EFNEB_UP FLWP FIWP EFRADL EFRADI SW_NEB RRTM_LW_NEB OTH_VIS OTH_NI1 OTH_NI2 OTH_NI3 SSA_VIS SSA_NI1 SSA_NI2 SSA_NI3 ASF_VIS ASF_NIR1 ASF_NIR2 ASF_NIR3 ODAER_VIS ODAER_NIR1 ODAER_NIR2 ODAER_NIR3 SSAAER_VIS SSAAER_NIR1 SSAAER_NIR2 SSAAER_NIR3 GAER_VIS GAER_NIR1 GAER_NIR2 GAER_NIR3
		O3CLIM: [3D] climatological ozone content (Pa/Pa)
		CUM_AER_LAND, CUM_AER_SEA CUM_AER_DES, CUM_AER_URB CUM_AER_VOL, CUM_AER_STRB [3D] cumulated optical thickness of the different aerosols from the top of the domain (_)

10.2.7 Lagrangian tracers

Only available if `LLG=T` in YINIFILE.des

LTRAJ	**.FALSE**	by default : no field
	.TRUE.	`X, Y` : [3D] X and Y coordinates (km)
		`LGXM, LGYM, LGZM` : [3D] Lagrangian tracers coordinates (m)
		`Xn, Yn, Zn` : [3D] Lagrangian tracers coordinates at time origin `n`
		`THn` : [3D] corresponding Theta (K)
		`RVn` : [3D] corresponding Vapor mixing Ratio (g/kg)

A documentation is available at http://mesonh.aero.obs-mip.fr/mesonh/dir_doc/lag_m46_22avril2005/lagrangian46/

10.2.8 Dust variables

Only available if `LDUST=T` in YINIFILE.des.

by default		`DSTM0nM`: [3D] Dust 0-order moment of the lognormal mode n (ppbv)
		`DSTM3nM` : [3D] Dust 3^{rd}-order moment of the lognormal mode n (ppbv)
		`DSTM6nM` : [3D] Dust 6^{rd}-order moment of mode n (if `LVARSIG`) (ppbv)
		`DSTRGAn` : [3D] Dust number mean Radius of the lognormal mode n (μm)
		`DSTRGAMn` : [3D] Dust Mass mean Radius of the lognormal mode n (μm)
		`DSTN0An` : [3D] Dust Number of the lognormal mode n (/m^3)
		`DSTSIGAn` : [3D] Dust Standard deviation of the lognormal mode n (_)
		`DSTMSSn` : [3D] Dust Mass concentration of the lognormal mode n (μg/m^3)
		`DSTBRDNn` : [2D] Dust Burden of the lognormal mode n (g/m^2)
		`DEDSTM3nCM` : [3D] Dust Mass of mode n in cloud water only if `LDEPOS_DST=T` (ppb)
		`DEDSTM3nRM` : [3D] Dust Mass of mode n in rain only if `LDEPOS_DST=T` (ppb)
		`DEDSTN0An` : [3D] Number of dust particles in cloud water (for n=1,2,3) or in rain (for n=4,5,6) only if `LDEPOS_DST=T` (/m^3)
		`DEDSTMSSn` : [3D] Dust mass in cloud water (for n=1,2,3) or in rain (for n=4,5,6) only if `LDEPOS_DST=T`(μg/m^3)
NRAD_3D	≥ 1	`DSTAOD2D`: [2D] Dust Optical Depth (_)
		`DSTAOD3D`: [3D] Dust Optical Depth between two vertical levels (_)
		`DSTEXT`: [3D] Dust Extinction (1/km)

10.2.9 Salt variables

Only available if `LSALT=T` in YINIFILE.des.

by default		`SLTM0nM`: [3D] Salt 0-order moment of the lognormal mode n (ppbv)
		`SLTM3nM` : [3D] Salt 3^{rd}-order moment of the lognormal mode n (ppbv)
		`SLTM6nM` : [3D] Salt 6^{rd}-order moment of mode n (if `LVARSIG_SLT`) (ppbv)
		`SLTRGAn` : [3D] Salt number mean Radius of the lognormal mode n (μm)
		`SLTRGAMn` : [3D] Salt Mass mean Radius of the lognormal mode n (μm)
		`SLTN0An` : [3D] Salt Number of the lognormal mode n (/m^3)
		`SLTSIGAn` : [3D] Salt Standard deviation of the lognormal mode n (_)
		`SLTMSSn` : [3D] Salt Mass concentration of the lognormal mode n (μg/m^3)
		`SLTBRDNn` : [2D] Salt Burden of the lognormal mode n (g/m^2)
		`DESLTM3nCM` : [3D] Salt Mass of mode n in cloud water only if `LDEPOS_SLT=T` (ppb)
		`DESLTM3nRM` : [3D] Salt Mass of mode n in rain only if `LDEPOS_SLT=T` (ppb)
		`DESLTN0An` : [3D] Number of salt particles in cloud water (for n=1,2,3) or in rain (for n=4,5,6) only if `LDEPOS_SLT=T` (/m^3)
		`DESLTMSSn` : [3D] Salt mass in cloud water (for n=1,2,3) or in rain (for n=4,5,6) only if `LDEPOS_SLT=T` (μg/m^3)
NRAD_3D	≥ 1	`DSTAOD2D`: [2D] Dust Optical Depth (_)
		`SLTAOD3D`: [3D] Salt Optical Depth between two vertical levels (_)
		`SLTEXT`: [3D] Salt Extinction (1/km)

10.2.10 Chemical variables

Only available if `LUSECHEM=T` in YINIFILE.des

LCHEMDIAG	**.FALSE.**	by default : no fields
	.TRUE.	`O3M ...` : [3D] Chemical scalar variables as defined in BASIC.f90 (ppbv)
XCHEMLAT, XCHEMLON	**XUNDEF**	by default : no fields
	arrays of real	write chemicals species on vertical profile defined by (XCHEMLAT,XCHEMLON)

10.2.11 Aerosol variables

Only available if `LUSECHEM=T` and `LORILAM=T` in YINIFILE.des

LCHEMDIAG	**.FALSE.**	by default : no field
	.TRUE.	`SOAIM ...` : [3D] Aerosol scalar variable as defined in ch_aer_init_soa.f90 (ppbv)
		`RGAn`: [3D] Aerosol number mean Radius of the lognormal mode n (μm)
		`RGAMn`: [3D] Aerosol Mass mean Radius of the lognormal mode n (μm)
		`NOAn`: [3D] Aerosol Number of the lognormal mode n (/cc)
		`SIGAn`: [3D] Aerosol Standard deviation of the lognormal mode n (_)
		`MSO4n`: [3D] Mass SO4 aerosol mode n (μm/m^3)
		`MNO3n`: [3D] Mass NO3 aerosol mode n (μm/m^3)
		`MNH3n`: [3D] Mass NH3 aerosol mode n (μm/m^3)
		`MH2On`: [3D] Mass H2O aerosol mode n (μm/m^3)
		`MSOA1n`: [3D] Mass SOA1 aerosol mode n (μm/m^3)
		`MSOA2n`: [3D] Mass SOA2 aerosol mode n (μm/m^3)
		`MSOA3n`: [3D] Mass SOA3 aerosol mode n (μm/m^3)
		`MSOA4n`: [3D] Mass SOA4 aerosol mode n (μm/m^3)
		`MSOA5n`: [3D] Mass SOA5 aerosol mode n (μm/m^3)
		`MSOA6n`: [3D] Mass SOA6 aerosol mode n (μm/m^3)
		`MSOA7n`: [3D] Mass SOA7 aerosol mode n (μm/m^3)
		`MSOA8n`: [3D] Mass SOA8 aerosol mode n (μm/m^3)
		`MSOA9n`: [3D] Mass SOA9 aerosol mode n (μm/m^3)
		`MSOA10n`: [3D] Mass SOA10 aerosol mode n (μm/m^3)
		`MOCn`: [3D] Mass OC aerosol mode n (μm/m^3)
		`MBCn`: [3D] Mass BC aerosol mode n (μm/m^3)

10.2.12 Production of NOx by lightening flashes

only available if LCH_CONV_LINOX=T and LUSECHEM=F in YINIFILE.des with LCHEMDIAG=F in DIAG1.nam

`LINOXM` : [3D] linox scalar variables
`IC_RATE` : [2D] IntraCloud lightning Rate (/s)
`CG_RATE` :[2D] CloudGround lightning Rate (/s)
`IC_TOTAL_NB` :[2D] IntraCloud lightning Number (_)
`CG_TOTAL_NB` :[2D] CloudGround lightning Number (_)

10.2.13 GPS synthetic delays

Fortran name	Fortran type	default value
CNAM_GPS	array(character)	50* ”
XLAT_GPS	array(real)	50* XUNDEF
XLON_GPS	array(real)	50* XUNDEF
XZS_GPS	array(real)	50* -999.
XDIFFORO	real	150.

- CNAM_GPS: name of the GPS stations
- XLAT_GPS: latitude of the GPS stations
- XLON_GPS: longitude of the GPS stations
- XZS_GPS: height of the GPS stations (m)
- XDIFFORO: maximum difference between model orography and station height accepted when computing interpolated delays value (m)

For stations where latitude, longitude and height are different from default values, the interpolated values of GPS delays are written in ASCII files *YINIFILEYSUFFIX*GPS[.P00n] (where n is the number of processor).

NGPS	**-1**	by default : no field
	0	`ZTD` : [2D] Zenithal Total Delay (m)
	1	`ZTD` : [2D] Zenithal Total Delay (m)
		`ZHD` : [2D] Zenithal Hydrostatic Delays (m)
		`ZWD` : [2D] Zenithal Wet Delays (m)

10.2.14 Computing Satellite image from a MESO-NH run

A comparison between model outputs and satellite observations provides an assessment on how well the model can reproduce the meteorological situation. The model-to-satellite approach compares directly the satellite brightness temperatures (BTs) to the BTs computed from the predicted model fields (Morcrette, 1991). It has been first applied to Meso-NH outputs for comparison with Meteosat observations in the infrared using a narrow-band code (Chaboureau et al., 2000).

CRAD_SAT	**' '**	by default : no computation is made
	'METEOSAT'	`METEOSAT_IRBT` : [2D] Brightness temperature in IR channel (K)
		`METEOSAT_WVBT` : [2D] Brightness temperature in WV channel (K)
	'GMS'	`GMS_IRBT` : [2D] Brightness temperature in IR channel (K)
		`GMS_WVBT` : [2D] Brightness temperature in WV channel (K)
	'GOES-E'	`GOES-E_IRBT` : [2D] Brightness temperature in IR channel (K)
		`GOES-E'_WVBT` : [2D] Brightness temperature in WV channel (K)
	'GOES-W'	`GOES-W_IRBT` : [2D] Brightness temperature in IR channel (K)
		`GOES-W_WVBT` : [2D] Brightness temperature in WV channel (K)
	'INDSAT'	`INDSAT_IRBT` : [2D] Brightness temperature in IR channel (K)
		`INDSAT_WVBT` : [2D] Brightness temperature in WV channel (K)
LRAD_SUBG_COND	**.TRUE.**	WITH subgrid condensation scheme taken into account
	.FALSE.	WITHOUT subgrid condensation scheme

Since the Masdev4_7 version, the Radiative Transfer for Tiros Operational Vertical Sounder (RTTOV) code version 8.7 (Saunders et al., 2005) is also available allowing the calculation of BT for a large number of satellites. **Use of RTTOV is highly recommended to compute brightness temperature.**

You need an additional file (300 Mo) which contains coefficients : `rttov87_rtcoef.tar`

NRTTOVinfo (1:4,nb)	**999**	by default : no computation is made
	Plt Sat Sen 0	`PltSatSenBT` : [2D] Brightness temperature (K) with $\begin{cases} Plt = \text{Plateforme} \\ Sat = \text{Satellite} \\ Sen = \text{Sensor} \\ nb = \text{number of instrument you want} \\ 1 \leq nb \leq 10 \end{cases}$ See below for more information

To simulate an instrument, use the code given in the following Tables reproduced from the RTTOV users guide (see http://www.metoffice.com/research/interproj/nwpsaf/rtm/rtm_rttov8.html). Platforms supported by RTTOV 8.7 at 17 Nov 2005 are in normal text. Platforms in italics are not yet supported by RTTOV 8.7 but soon will be.

Platform	*Plt*	*Sat* range
NOAA	1	1 to 18
DMSP	2	8 to 16
Meteosat	3	5 to 7
GOES	4	8 to 12
GMS	5	5
FY-2	6	2 to 3
TRMM	7	1
ERS	8	1 to 2
EOS	9	1 to 2
METOP	*10*	*1 to 3*
ENVISAT	11	1
MSG	12	1 to 2
FY-1	13	3
ADEOS	14	1 to 2
MTSAT	15	1
CORIOLIS	16	1

Sensor	RTTOVid (*Sen*)	Sensor Channel #	RTTOV-8 Channel #
HIRS	0	1 to 19	1 to 19
MSU	1	1 to 4	1 to 4
SSU	2	1 to 3	1 to 3
AMSU-A	3	1 to 15	1 to 15
AMSU-B	4	1 to 5	1 to 5
AVHRR	5	3b to 5	1 to 3
SSMI	6	1 to 7	1 to 4
VTPR1	7	1 to 8	1 to 8
VTPR2	8	1 to 8	1 to 8
TMI	9	1 to 9	1 to 9
SSMIS	10	1 to 24	1 to 21
AIRS	11	1 to 2378	1 to 2378
HSB	12	1 to 4	1 to 4
MODIS	13	1 to 17	1 to 17
ATSR	14	1 to 3	1 to 3
MHS	15	1 to 5	1 to 5
IASI	16	1 to 8461	1 to 8461
AMSR	17	1 to 14	1 to 7
MVIRI	20	1 to 2	1 to 2
SEVIRI	21	4 to 11	1 to 8
GOES-Imager	22	1 to 4	1 to 4
GOES-Sounder	23	1 to 18	1 to 18
GMS/MTSAT imager	24	1 to 4	1 to 4
FY2-VISSR	25	1 to 2	1 to 2
FY1-MVISR	26	1 to 3	1 to 3
CriS	27	TBD	TBD
CMISS	28	TBD	TBD
VIIRS	29	TBD	TBD
WINDSAT	30	1 to 10	1 to 5

10.2.15 Radar

<table>
<tr><td rowspan="6">LRADAR=T</td><td>.FALSE.</td><td colspan="2">by default : no field</td></tr>
<tr><td rowspan="5">.TRUE.</td><td rowspan="4">NVERSION_RAD=1</td><td>RARE : [3D] (dBZ)
Radar Reflectivity</td></tr>
<tr><td>VDOP : [3D] (m/s)
radar Doppler fall speed</td></tr>
<tr><td>ZDR : [3D] (dBZ)
radar Differential Reflectivity</td></tr>
<tr><td>KDP : [3D] (degree/km)
radar Differential Phase shift</td></tr>
<tr><td>NVERSION_RAD=2</td><td>Simulator of radar
See below for more informations</td></tr>
</table>

Simulator of Radar

A radar simulator already existed in Meso-NH (Richard et al.,2003) that computes reflectivities in the Rayleigh approximation on each grid points of the model : (NVERSION=1). However, with the view to code an observation operator for radar reflectivities, this simulator was not sufficient. That is why a new simulator was built, while the original version is still available. This new simulator (NVERSION=2) simulates Plan Position Indicators (PPI), which are cones usually projected on a horizontal plane obtained by scanning the atmosphere at constant elevation. New features are:

- possibility to choose among several scattering models,
- beam bending taken into account,
- possibility to take attenuation into account,
- antenna's radiation pattern (beam broadening) modeled,
- ouptut on operational (Cartesian) grids of the Aramis French radar network.

Fortran name	Fortran type	Values	Meaning
`XLAT_RAD`	array of reals	XUNDEF	latitude of each radar
`XLON_RAD`	array of reals	`XUNDEF`	longitude of each radar
`XALT_RAD`	array of reals	`XUNDEF`	altitudes of radars (m)
`CNAME_RAD`	array of strings	`"XUNDEF"`	names of radars
`XLAM_RAD`	array of reals	`XUNDEF`	radar wavelengths
`XDT_RAD`	array of reals	`XUNDEF`	beam width to the -3 dB level for one-way transmission ($\Delta\theta$
`XELEV`	2-dim array of reals	`XUNDEF`	radar elevations (θ, in $^\circ$). First dimension: radar; second: site number
`NBSTEPMAX`	integer	-1	number of gates
`XSTEP_RAD`	real	`XUNDEF`	gate length (m)
`LATT`	logical	`.FALSE.`	attenuation is taken into account if true
`LQUAD`	logical	`.FALSE.`	if true Gauss-Legendre quadrature if false Gauss-Hermite quadrature
`NPTS_H`	integer	1	number of angles for the quadrature in horizontal
`NPTS_V`	integer	1	number of angles for the quadrature in vertical
`CARF`	string	`"PB70"`	(default) axis ratio of raindrops : Pruppacher and Beard (1970)
		`"AND99"`	axis ratio of raindrops : Andsager et al. (1999).
`LREFR`	logical	`.FALSE.`	if true writes out refractivity ($N \equiv (n-1) \times 10^6$)
`LDNDZ`	logical	`.FALSE.`	if true writes out vertical gradient of refractivity (Ṇ/ẓ)
`NCURV_INTERPOL`	integer	0	use an average beam bending equivalent to 4/3 of the Earth's radius
		1	compute the beam bending at each gate by using model variables
`LCART_RAD`	logical	`.TRUE.`	true if interpolation of reflectivity on a cartesian grid ; false if polar
`NDIFF`	integer	0	Rayleigh scattering
		1	Mie scattering
		2	T-matrix scattering
		3	Rayleigh for spheroids scattering
		4	Rayleigh with 6^{th} order for attenuation calculations
`NPTS_GAULAG`	integer	7	number of points of the quadrature
`XGRID`	real	2000.	size of the Cartesian grid (m)
`LFALL`	logical	`.FALSE.`	if true takes into account hydrometeor fall speeds
`LWREFL`	logical	`.FALSE.`	if true takes into account the weighting by reflectivities
`LWBSCS`	logical	`.FALSE.`	if true takes into account the weighting by hydrometeor concentrations
`XREFLMIN`	real	$-30.$	minimum detectable reflectivity (in dBZ)
`XREFLVDOPMIN`	real	$-990.$	minimum detectable reflectivity to compute Doppler velocities (in dBZ; useless when `LWREFL=.FALSE.`)

Output files

As output fields are not on the model grid, they have to be written in other files than LFIs. Therefore the following files are written in the following format: AAABBBCC.CDDD**X**, where AAA is the descriptor of the field (3 characters, see below for further explanations), BBB is the name of the radar (3 characters), CC.C is the elevation (in degrees), DDD is half the number of pixels on each row or column (3 characters), and **X** is the name of the input file. Example of file name: `ZHHBOL00.4300BOG12.2.SEG04.004RD`.

Field descriptors can be

- ZHH : overall reflectivity (dBZ),
- ZER : reflectivity due to rain (dBZ),
- ZEI : reflectivity due to pristine ice (dBZ),
- ZES : reflectivity due to snow (dBZ),
- ZEG : reflectivity due to graupel (dBZ),
- KDP : specific differential phase ($^\circ km^{-1}$),
- ZDR : differential reflectivity (dB),
- VRU : Doppler velocity (m s^{-1}),
- HAS : height of middle of beam MSL (m),
- M_R : rainwater contents in the middle of the beam (kg kg^{-1}),
- M_I : primary ice contents in the middle of the beam (kg kg^{-1}),
- M_S : snow contents in the middle of the beam (kg kg^{-1}),
- M_G : graupel contents in the middle of the beam (kg kg^{-1}),
- CIT : pristine ice concentration in the middle of the beam (kg m^{-3}),
- AET : overall two-way specific attenuation (dB km^{-1}) (if LATT=T),
- AER : two-way specific attenuation due to rain (dB km^{-1}) (if LATT=T),
- AEI : two-way specific attenuation due to pristine ice (dB km^{-1}) (if LATT=T),
- AES : two-way specific attenuation due to snow (dB km^{-1}) (if LATT=T),
- AEG : two-way specific attenuation due to graupel (dB km^{-1}) (if LATT=T),
- ATT : overall two-way path-integrated attenuation (PIA) (dB) (if LATT=T),
- ATR : two-way PIA due to rain (dB) (if LATT=T),
- ATI : two-way PIA due to pristine ice (dB) (if LATT=T),
- ATS : two-way PIA due to snow (dB) (if LATT=T),
- ATG : two-way PIA due to graupel (dB) (if LATT=T),
- RFR : refractivity in the middle of the beam (if LREFR=T),

- DNZ : vertical gradient of refractivity in the middle of the beam (km^{-1}) (if LDNDZ=T),
- CSR : index characterizing the pixel: 0 stands for clear-air, 1 for stratiform, 2 for convective.

10.2.16 Lidar

<table>
<tr><td rowspan="3">LLIDAR</td><td>.FALSE.</td><td>by default : no field</td></tr>
<tr><td rowspan="2">.TRUE</td><td>LIDAR : [3D] total backscatter coefficient (1/m/sr)</td></tr>
<tr><td>LIPAR : [3D] particle backscatter coefficient (1/m/sr)</td></tr>
</table>

Fortran name	Fortran type	default value
CVIEW_LIDAR	character(*5)	'NADIR'
XALT_LIDAR	real	0
XWVL_LIDAR	real	0.532E-6

- CVIEW_LIDAR : gives the lidar point of view : either 'NADIR' or 'ZENIT'
- XALT_LIDAR : gives the altitude of the lidar source (in meters) (by default, the altitude of the ground will be used for zenithal view, and the altitude of the model top will be used for nadir view)
- XWVL_LIDAR : gives the wavelength of the lidar source (in meters)

10.2.17 Aircraft and balloon

Fortran name	Fortran type	default value
LAIRCRAFT_BALLOON	logical	.FALSE.
NTIME_AIRCRAFT_BALLOON	integer	NUNDEF
XSTEP_AIRCRAFT_BALLOON	real	XUNDEF
XLAT_BALLOON	array(real)	9*XUNDEF
XLON_BALLOON	array(real)	9*XUNDEF
XALT_BALLOON	array(real)	9*XUNDEF

- LAIRCRAFT_BALLOON : flag to compute aircraft and balloon trajectories with stationary fields. Trajectories will be written in diachronic file *YINIFILE*BAL
- NTIME_AIRCRAFT_BALLOON: time length of trajectories computation centered on CURrent time (s)
- XSTEP_AIRCRAFT_BALLOON: minimum time step for trajectories computation (s)
- XLAT_BALLOON: initial latitudes of the balloons
- XLON_BALLOON: initial longitudes of the balloons
- XALT_BALLOON: initial altitudes of the balloons (m)

10.3 Externalized surface diagnostics

Here are listed the namelists of SURFEX used for DIAG step. See SURFEX documentation for details.

- NAM_DIAG_SURF_ATMn
- NAM_WRITE_DIAG_SURFn
- NAM_DIAG_SURFn
- NAM_DIAG_ISBAn
- NAM_DIAG_TEBn
- NAM_DIAG_FLAKEn
- NAM_DIAG_OCEANn

10.4 Examples of DIAG1.nam

- Example 1

```
&NAM_DIAG
  LVAR_LS=T, NCONV_KF=2, NRAD_3D=1,
  CRAD_SAT='METEOSAT',
  LVAR_MRW=T, LVAR_MRSV=T, LMOIST_V=T, LMOIST_E=F,
  LTPZH=T, LVORT=F, LMSLP=F, LGEO=T, LAGEO=T, LWIND_ZM=F,
  LTHW=T, LCLD_COV=T,
  LVAR_PR=F, LTOTAL_PR=F, LMEAN_PR=F, XMEAN_PR(1,2)=4. ,
  NCAPE=1, LRADAR=T, LTRAJ=F /
&NAM_DIAG_BLANK /
&NAM_DIAG_FILE  YINIFILE(1) = "F9801.1.06A12.002" ,
                YSUFFIX = "diag" /
&NAM_DIAG_ISBAn N2M=2, LSURF_BUDGET=T /
```

- Example 2 : Namelist file for 6 files using trajectories computation

```
&NAM_DIAG
  LVAR_PR=T, LTOTAL_PR=T, LTPZH=T, LVAR_MRSV=T, LTRAJ=T /
&NAM_DIAG_FILE  YSUFFIX='d18-6'
                YINIFILE(1) = "NAPE2.1.APE05.001" ,
&NAM_STO_FILE   CFILES(1) = "NAPE2.1.APE05.001" ,
                CFILES(2) = "NAPE2.1.APE04.001" ,
                CFILES(3) = "NAPE2.1.APE03.001" ,
                CFILES(4) = "NAPE2.1.APE02.001" ,
                CFILES(5) = "NAPE2.1.APE01.001" ,
                CFILES(6) = "APE10_ARP19990919.18" ,
                NSTART_SUPP(1)= 4 ,
                NSTART_SUPP(2)= 2 /
```

- Example 3 : Namelist file for simulator of radar To simulate the radar of Nancy, with T-matrix scattering, for 1 elevation (1.3°)

```
 &NAM_DIAG
  LRADAR=T,NVERSION_RAD=2,NCURV_INTERPOL=0,LCART_RAD=T,
  LQUAD=F,LWBSCS=T,LDNDZ=F, LFALL=F,LWREFL=F,LREFR=F,
  NPTS_GAULAG=7,NPTS_H=1,NPTS_V=1,CARF="AND99",
```

```
  NDIFF=2,NBSTEPMAX=400,XSTEP_RAD=700.,XGRID=2000.,LATT=F,
  XELEV(1,1)=01.3, XLAT_RAD(1)=48.7167,XLON_RAD(1)=6.5825,XALT_RAD(1)=297.55,
  CNAME_RAD(1)="NANCY",XLAM_RAD(1)=0.0535,XDT_RAD(1)=1.3
/
&NAM_DIAG_FILE  YSUFFIX = "RD",YINIFILE = "ALD00.2.SOG12.004" /
```

Chapter 11

Compute spectra after a MESO-NH simulation

11.1 Presentation

After running the model, you can compute spectra with the program SPECTRE, that gives the kinetic energy density according to the wavelength or the wave number (see Ricard et al., 2011). The calculation uses a discrete cosinus transform to convert grid-point fields into spectral space ones.

11.1.1 Input file

The "SPECTRE" program computes spectra on MESONH files issued from a simulation or on AROME file. In this second case (CTYPEFILE='AROME'), the program does not read the true AROME file but an ASCII file obtained with the tools "edf". You must create one ASCII file per variable you want : the files will be named YINIFILE_U, YINIFILE_V...

11.1.2 Output files

You will obtain one ascii file per variable you have asked. A script exists to trace easily the results. (on MESONH web site in Wiki Team's)

11.2 The namelist file SPEC1.nam

11.2.1 Namelist NAM_SPECTRE_FILE

Fortran name	Fortran type	default value
YINIFILE	array of character (len=28)	' '
CTYPEFILE	character (LEN=6)	'MESONH'
YOUTFILE	array of character (len=28)	' '
LSTAT	logical	.FALSE.

- YINIFILE : name of the input FM file.

- CTYPEFILE : type of the input file ('AROME ', 'MESONH')
- YOUTFILE : prefix of the output file.
  If the user does specify this name, the output file will be named YOUTFILE_U, YOUTFILE_V ....
  If the user does NOT specify this name, the output file will be named spectra_YINIFILE_U, spectra_YINIFILE_V ....
- LSTAT : flag to have some statistiques on fields if .TRUE.

11.2.2 Namelist NAM_SPECTRE

Fortran name	Fortran type	default value
LSPECTRE_U	logical	.FALSE.
LSPECTRE_V	logical	.FALSE.
LSPECTRE_W	logical	.FALSE.
LSPECTRE_TH	logical	.FALSE.
LSPECTRE_RV	logical	.FALSE.
LSPECTRE_LSU	logical	.FALSE.
LSPECTRE_LSV	logical	.FALSE.
LSPECTRE_LSW	logical	.FALSE.
LSPECTRE_LSTH	logical	.FALSE.
LSPECTRE_LSRV	logical	.FALSE.
LSMOOTH	logical	.FALSE.

- LSPECTRE_U : flag to compute spectrum of U component
- LSPECTRE_V : flag to compute spectrum of V component
- LSPECTRE_W : flag to compute spectrum of W
- LSPECTRE_TH : flag to compute spectrum of Theta
- LSPECTRE_RV : flag to compute spectrum of vapor mixing ratio
- LSPECTRE_LSU : flag to compute spectrum of large scale U component
- LSPECTRE_LSV : flag to compute spectrum of large scale V component
- LSPECTRE_LSW : flag to compute spectrum of large scale W
- LSPECTRE_LSTH : flag to compute spectrum of large scale Theta
- LSPECTRE_LSRV : flag to compute spectrum of large scale vapor mixing ratio
- LSMOOTH : flag to smooth the results

11.2.3 Namelist NAM_ZOOM_SPECTRE

Fortran name	Fortran type	default value
LZOOM	logical	.FALSE.
NXDEB	integer	
NYDEB	integer	
NITOT	integer	
NJTOT	integer	

- LZOOM : flag to make a zoom on the file domain
- NXDEB : first point I index, left to and out of the wanted domain
- NYDEB : first point J index, under and out of the wanted domain
- NITOT : number of grid points in I direction. It must be equal to $2^m \times 3^n \times 5^p$ with $(m, n, p) \geq 0$
- NJTOT : number of grid points in J direction. It must be equal to $2^m \times 3^n \times 5^p$ with $(m, n, p) \geq 0$

11.2.4 Namelist NAM_DOMAIN_AROME

This namelist is only available for CTYPEFILE='AROME'. It contains the caracteristics of the domain arome in the input file.

Fortran name	Fortran type	default value
NI	integer	750
NJ	integer	720
NK	integer	60
XDELTAX	integer	2500
XDELTAY	integer	2500

- NI : number of points to read in I direction
- NJ : number of points to read in J direction
- NK : : number of vertical levels to read
- XDELTAX : gridsize of arome file in I direction (m)
- XDELTAY : gridsize of arome file in J direction (m)

Appendix A

Name of the variables in MESONH

We will make a list of the variables present in a MESONH file without LES and budget variables. For the DIAG program the list is made in the chapter 10. Only the MESONH variables are referenced, not SURFEX one.

Name	Dim	Meaning	Unit
ACPRC	[D]	Accumulated Cloud Precipitation Rain Rate	mm/h
ACPRG	[D]	Accumulated Precipitation Graupel Rate	mm/h
ACPRH	[D]	Accumulated Precipitation Hail Rate	mm/h
ACPRR	[D]	Accumulated Precipitation Rain Rate	mm/h
ACPRS	[D]	Accumulated Precipitation Snow Rate	mm/h
ACPRT	[D]	Total Accumulated Precipitation Rate	mm/h
AZIM	[2D]	azimuth	rad
CG_RATE	[2D]	CloudGround lightning Rate	/s
CG_TOTAL_NB	[2D]	CloudGround lightning Number	-
CLDFR	[2D]	Cloud fraction	
CLEARCOL_TM1	[2D]	Trace of cloud	-
DIR_ALB	[2D]	Direct albedo	-
DIRFLASWD	[2D]	Direct Downward Long Waves on flat surface	W/m^2
DIRSRFSWD	[2D]	Direct Downward Long Waves	W/m^2
DSVCONVxxx	[3D]	Convective tendency for scalar variable	/s
DSVCONV_LINOX	[3D]	Convective tendency for linox	/s
DRCCONV	[2D]	Convective R_c tendency	/s
DRICONV	[2D]	Convective R_i tendency	/s
DRVCONV	[2D]	Convective R_v tendency	/s
DTHCONV	[2D]	Convective heating/cooling rate	K/s
DTHRAD	[2D]	Radiative heating/cooling rate	K/s

Name	Dim	Meaning	Unit
EMIS	[2D]	Emissivity	-
EVAP3D	[2D]	Instantaneous 3D Rain Evaporation flux	kg/kg/s
EXNTOP		Exner function at model top	
FLALWD	[2D]	Downward Long Waves on flat surface	W/m^2
GXTHFRC	[1D]	$(\partial\theta/\partial x)_{frc}$	K/m
GYTHFRC	[1D]	$(\partial\theta/\partial y)_{frc}$	K/m
IC_RATE	[2D]	IntraCloud lightning Rate	/s
IC_TOTAL_NB	[2D]	IntraCloud lightning Number	-
INPRC	[2D]	Instantaneous Cloud Precipitation Rain Rate	mm/h
INPRG	[2D]	Instantaneous Precipitation Graupel Rate	mm/h
INPRH	[2D]	Instantaneous Precipitation Hail Rate	mm/h
INPRR	[2D]	Instantaneous Precipitation Rain Rate	mm/h
INPRR3D	[2D]	Instantaneous 3D Rain Precipitation flux	m/s
INPRS	[2D]	Instantaneous Precipitation Snow Rate	mm/h
INPRT	[2D]	Total Instantaneous Precipitation Rate	mm/h
LSPABSM	[3D]	Large scale absolute pression at $t - dt$ time	Pa
LSPABST	[3D]	Large scale absolute pression at t time	Pa
LSRVM	[3D]	Large scale Vapor mixing Ratio at $t - dt$ time	kg/kg
LSRVT	[3D]	Large scale Vapor mixing Ratio at t time	kg/kg
LSTHM	[3D]	Large scale potential temperature at $t - dt$ time	K
LSTHT	[3D]	Large scale potential temperature at t time	K
LSUM	[3D]	Large scale horizontal component U of wind at $t - dt$ time	m/s
LSUT	[3D]	Large scale horizontal component U of wind at t time	m/s
LSVM	[3D]	Large scale horizontal component V of wind at $t - dt$ time	m/s
LSVT	[3D]	Large scale horizontal component V of wind at t time	m/s
LSWM	[3D]	Large scale vertical wind at $t - dt$ time	m/s
LSWT	[3D]	Large scale vertical wind at t time	m/s
PABSM	[3D]	absolute pression at $t - dt$ time	Pa
PABST	[3D]	absolute pression at t time	Pa
PACCONV	[2D]	Convective Accumulated Precipitation rate (from the beginnning of the experiment)	mm
PGROUNDFRC	[0D]	forcing ground pressure	Pa
PRCONV	[2D]	Convective instantaneous Precipitation Rate	mm/h

Name	Dim	Meaning	Unit
PRSCONV	[2D]	Convective instantaneous Precipitation Rate for Snow	mm/h
RCM	[3D]	Cloud mixing Ratio at $t - dt$ time	kg/kg
RCT	[3D]	Cloud mixing Ratio at t time	kg/kg
RGM	[3D]	Graupel mixing Ratio at $t - dt$ time	kg/kg
RGT	[3D]	Graupel mixing Ratio at t time	kg/kg
RHM	[3D]	Hail mixing Ratio at $t - dt$ time	kg/kg
RHODREF	[3D]	Dry density for reference state with orography	kg/m^3
RHOREFZ	[1D]	rhodz for reference state without orography	kg/m^3
RHT	[3D]	Hail mixing Ratio at t time	kg/kg
RIM	[3D]	Ice mixing Ratio at $t - dt$ time	kg/kg
RIT	[3D]	Ice mixing Ratio at t time	kg/kg
RRM	[3D]	Rain mixing Ratio at $t - dt$ time	kg/kg
RRT	[3D]	Rain mixing Ratio at t time	kg/kg
RSM	[3D]	Snow mixing Ratio at $t - dt$ time	kg/kg
RST	[3D]	Snow mixing Ratio at t time	kg/kg
RVFRC	[1D]	$(\partial r_v/\partial t)_{frc}$ forcing vapor mixing ratio	kg/kg
RVM	[3D]	Vapor mixing Ratio at $t - dt$ time	kg/kg
RVT	[3D]	Vapor mixing Ratio at t time	kg/kg
SCA_ALB	[2D]	Scattered albedo	-
SCAFLASWD	[2D]	Scattered Downward Long Waves on flat surface	W/m^2
SVMnnn	[3D]	User or passive scalar variables at $t - dt$ time	kg/kg
SVTnnn	[3D]	User or passive scalar variables at t time	kg/kg
TENDRVFRC	[1D]	$(\partial r_v/\partial t)_{frc}$	/s
TENDTHFRC	[1D]	$(\partial \theta/\partial t)_{frc}$	K/s
THFRC	[1D]	θ_{frc} forcing potential temperature	K
THM	[3D]	potential temperature at $t - dt$ time	K
THT	[3D]	potential temperature at t time	K
THVREF	[3D]	Thetav for reference state with orography	K
THVREFZ	[1D]	thetavz for reference state without orography	K
TKEM	[3D]	Turbulent Kinetic Energy at $t - dt$ time	m^2/s^2
TKET	[3D]	Turbulent Kinetic Energy at t time	m^2/s^2
TSRAD	[2D]	Radiative Surface Temperature	K
UFRC	[1D]	zonal component of horizontal forcing wind	m/s
UM	[3D]	horizontal component U of wind at $t - dt$ time	m/s
UT	[3D]	horizontal component U of wind at t time	m/s
VFRC	[1D]	meridian component of horizontal forcing wind	m/s
VM	[3D]	horizontal component V of wind at $t - dt$ time	m/s
VT	[3D]	horizontal component V of wind at t time	m/s
WFRC	[D]	vertical forcing wind	m/s
WM	[3D]	vertical wind at $t - dt$ time	m/s
WT	[3D]	vertical wind at t time	m/s
ZENITH	[2D]	zenith	rad
ZS	[2D]	orography	m
ZSMT	[2D]	smoothed orography for SLEVE vertical coordinate	m

Hurricane initialization in PREP_REAL_CASE program

Name	Dim	Meaning	Unit
`UT15`	[3D]	component U of Total wind	m/s
`VT15`	[3D]	component V of Total wind	m/s
`TEMPTOT`	[3D]	Total Temperature	K
`PRESTOT`	[3D]	Total pressure	Pa
`UT16`	[3D]	component U of Environmental wind	m/s
`VT16`	[3D]	component V of Environmental wind	m/s
`TEMPENV`	[3D]	Environmental Temperature	K
`PRESENV`	[3D]	Environmental pressure	Pa
`UT17`	[3D]	component U of Basic (filtered) wind	m/s
`VT17`	[3D]	component V of Basic (filtered) wind	m/s
`TEMPBAS`	[3D]	Basic (filtered) Temperature	K
`PRESBAS`	[3D]	Basic (filtered) pressure	Pa
`VTDIS`	[3D]	Total disturbance tangential wind	m/s

Appendix B

Example of initialisation sequence for grid-nesting run

- The following initialisation and gridnesting sequence is shown here for three models, model 2 included in model 1, and model 3 included in model 2 (figure (B.1)):

  1. **PREP_PGD**: this program is run as many time as the number of models:
     - one physiographic data file for the model 1 (definition of projection, resolution, domain)
     - one physiographic data file for the model 2 (same projection, definition of resolution, domain)
     - one physiographic data file for the model 3 (same projection, definition of resolution, domain)
  2. **PREP_NEST_PGD**: this program checks all the three PGD files at the same time, and imposes the conformity between them.
  3. **extractecmwf** or **extractarpege**: it extracts the surface and altitude fields for one date, for model 1. The extraction must be done separately for each date and time (for the initial file and each of the coupling file of model 1).
  4. **PREP_REAL_CASE**: this program is running several times, for the initial file and the coupling files of model 1.
  5. **MESONH**: this step is **optional**. If you do not wish to start all the models at the same time, you can decide to run the model 1 before the model 2 starts.
  6. **ZOOM_PGD**: this step is **optional**. If you want to start the model 2 on a smaller domain than the one of the PGD file defined at steps 1 and 2 for the model 2, you must use this program.
  7. **SPAWNING**: when you want to start the model 2, you must use this program to compute the horizontal interpolations from the model 1 to the model 2. It is used

only once for the initialisation of model 2.

8. **PREP_REAL_CASE**: It is used only once, to compute the initial file for the model 2. **Do not change the vertical grid**.
9. **MESONH**: again, this step is **optional**. If you do not wish to start model 3 at the same time as model 2, you can decide to run the models 1 and 2 alone before.
10. **ZOOM_PGD**: again, this step is **optional**. If you want to start the model 3 on a smaller domain than the one of the PGD file defined at steps 1 and 2 for the model 3, you must use this program.
11. **SPAWNING**: when you want to start the model 3, you must use this program to compute the horizontal interpolations from the model 2 to the model 3. It is used only once for the initialisation of model 3.
12. **PREP_REAL_CASE**: It is used only once, to compute the initial file for the model 3. **Do not change the vertical grid**.
13. **MESONH**: here is now your complete nested run.

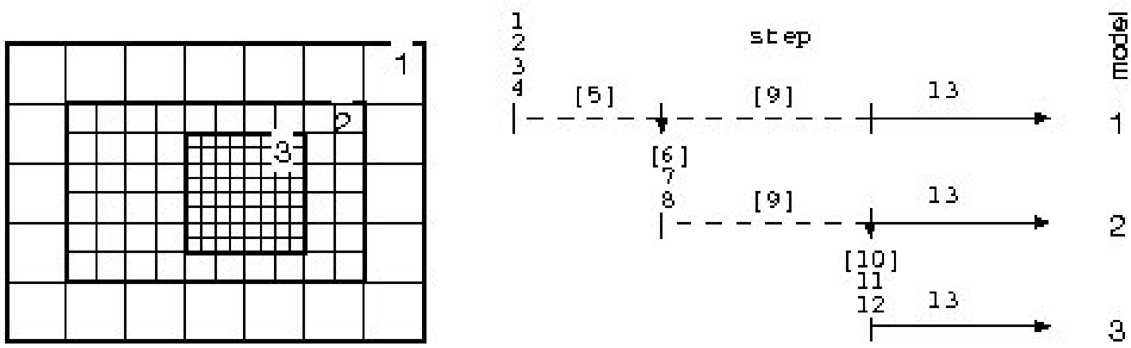


Figure B.1: Exemple of a grid-nesting simulation with 3 nested models

- The following initialisation and gridnesting sequence is shown here for three models, model 2 included in model 1, and model 3 included in model 1. Model 3 has the same resolution as model 2 and is started after model 2 to follow atmospheric system (figure (B.2)):

  1. **PREP_PGD**:
     - one physiographic data file for the model 1 (definition of projection, resolution, domain)
     - one physiographic data file for the models 2 and 3 (same projection, definition of resolution, domain)
  2. **PREP_NEST_PGD**: this program checks all the two PGD files at the same time, and imposes the conformity between them.
  3. **extractecmwf** or **extractarpege**: it extracts the surface and altitude fields for one date, for model 1. The extraction must be done separately for each date and time (for the initial file and each of the coupling file of model 1).
  4. **PREP_REAL_CASE**: this program is run several times, for the initial file and the coupling files of model 1.
  5. **MESONH**: this step is **optional**. If you do not wish to start all the models at the same time, you can decide to run the model 1 before the model 2 starts.
  6. **ZOOM_PGD**: Since the second PGD file was done for models 2 and 3, you have to zoom it on the domain of model 2 with this program.
  7. **SPAWNING**: when you want to start the model 2, you must use this program to compute the horizontal interpolations from the model 1 to the model 2. It is used only once for the initialisation of model 2.
  8. **PREP_REAL_CASE**: It is used only once, to compute the initial file for the model 2. **Do not change the vertical grid**.
  9. **MESONH**: here is your complete nested run with model 1 and model 2.
  10. **ZOOM_PGD**: Since the second PGD file was done for the models 2 and 3, you have to zoom it on the domain of model 3 with this program. The domain of model 3 has a common zone with the one of model 2.
  11. **SPAWNING**: when you want to start the model 3, you must use this program to compute the horizontal interpolations from the model 1 and to use the fields of model 2 in the common domain. It is used only once for the initialisation of model 3.
  12. **PREP_REAL_CASE**: It is used only once, to compute the initial file for the model 3. **Do not change the vertical grid**.

13. **MESONH**: here is the nested run with model 1 and model 3.

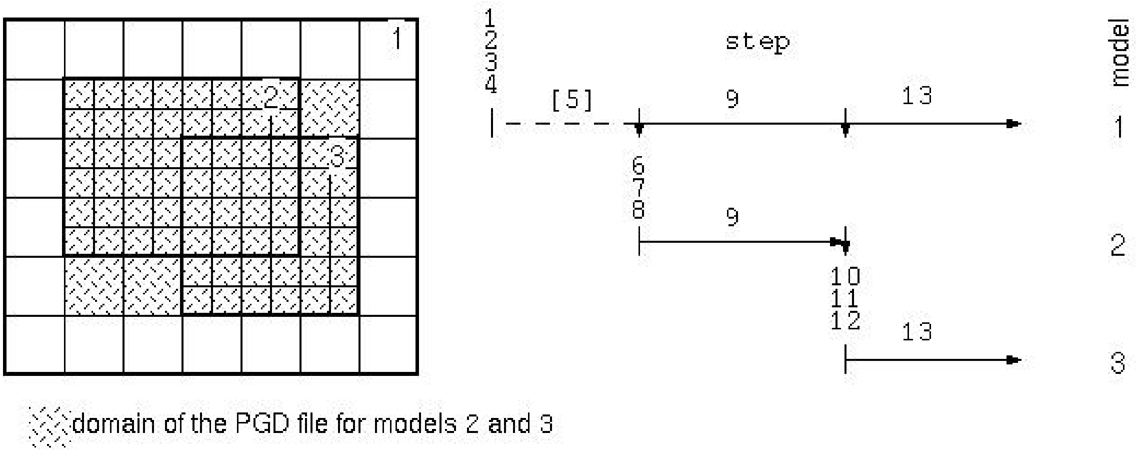


Figure B.2: Exemple of a grid-nesting simulation with 3 nested models (the domain of the 2 finest models has the same resolution and a common zone to follow atmospheric system).

Appendix C

LES diagnostics

C.1 Notations

α'	subgrid fluctuation of α	
$\overline{\alpha}$	mean value of α in the grid: resolved quantity	3D
$< \alpha >$	horizontal mean value of α	1D
$\tilde{\alpha} = \overline{\alpha} - < \alpha >$	resolved fluctuation to the mean profile	3D
$< \alpha >_{up}$	horizontal mean value of α in updrafts; only point with $\overline{w}$ greater than $< w >$ are considered	1D
$< \alpha >_{down}$	horizontal mean value of α in downdrafts; only point with $\overline{w}$ smaller than $< w >$ are considered	1D

Examples:

$\alpha'\beta'$		subgrid flux or (co)variance $\alpha'\beta'$	3D
$\overline{\alpha}$		mean value of α in each grid mesh	3D
$\overline{\alpha'}$	$= 0$	mean value of the turbulent fluctuation in each grid mesh	3D
$\overline{\alpha'\beta'}$		mean value in each grid mesh of subgrid flux or (co)variance	3D
$\tilde{\alpha}\,\tilde{\beta}$		resolved flux or (co)variance in each grid mesh	3D
$< \alpha >$		horizontal mean value of α	1D
$< \alpha' >$	$= 0$	horizontal mean value of a subgrid fluctuation	1D
$< \overline{\alpha} >$	$=< \alpha >$	horizontal mean value of a resolved field	1D
$< \tilde{\alpha} >$	$= 0$	horizontal mean value of a resolved fluctuation	1D
$< \overline{\alpha'\beta'} >$		horizontal mean value of subgrid flux or (co)variance	1D
$< \tilde{\alpha}\,\tilde{\beta} >$		horizontal mean value of resolved flux or (co)variance	1D

C.2 What is available

The computed fields have usually at least two dimensions: z and t, that is they are temporal evolutions of vertical profiles. They are always written in the diachronic file. Each field have its own group name, say 'NAME'.

When time averaging is asked for, the fields are temporally averaged (and so lose their temporal dimension) , and are written under the name 'A_NAME'.

When normalization is asked for, this one is made individually on each vertical profile, for all times. They are written under the name 'E_NAME'.

When both normalization and time averaging are asked for, normalization is made first, and then time averaging. The resulting vertical profiles are written under the name 'H_NAME'.

C.3 LES averaged fields (LLES_MEAN=TRUE)

field	notation in the diachronic file	dim.	if	comments
$< u >$	MEAN_U	z,t,p		dimension 'p' is equal to the number of masks when this dimension is not present, the computation is made only on the cartesian mask.
$< v >$	MEAN_V	z,t,p		
$< w >$	MEAN_W	z,t,p		
$< p >$	MEAN_PRE	z,t,p		
$< \rho >$	MEAN_RHO	z,t,p		
$< \theta >$	MEAN_TH	z,t,p		
$< \theta_l >$	MEAN_THL	z,t,p	r_c	
$< \theta_v >$	MEAN_THV	z,t,p	r_v	
$< r_v >$	MEAN_RV	z,t,p	r_v	
$< r_c >$	MEAN_RC	z,t,p	r_c	
$< r_r >$	MEAN_RR	z,t,p	r_r	
$< r_i >$	MEAN_RI	z,t,p	r_i	
$< r_s >$	MEAN_RS	z,t,p	r_s	
$< r_g >$	MEAN_RG	z,t,p	r_g	
$< r_h >$	MEAN_RH	z,t,p	r_h	
$< s_v >$	MEAN_SV	z,t,p,n	s_v	
$< \sqrt{\overline{u}^2 + \overline{v}^2} >$	MEAN_WIND	z,t,p		different from $\sqrt{< \overline{u} >^2 + < \overline{v} >^2}$!
$< \overline{\rho} \max(\overline{w}, < w >) >$	MEAN_MSFX	z,t,p		mean upward mass flux

C.4 LES pdf (LLES_PDF=TRUE)

field	notation in the diachronic file	dim.	if	comments
PDF_θ	PDF_TH	z,t,p,n		dimension 'p' is equal to the number of masks when this dimension is not present, the computation is made only on the cartesian mask.
PDF_W	PDF_W	z,t,p,n		
PDF_θ_v	PDF_THV	z,t,p,n		
PDF_R_v	PDF_RV	z,t,p,n		
PDF_R_c	PDF_RC	z,t,p,n		
PDF_R_t	PDF_RT	z,t,p,n		
PDF_θ_l	PDF_THL	z,t,p,n		
PDF_R_r	PDF_RR	z,t,p,n		
PDF_R_i	PDF_RI	z,t,p,n		
PDF_R_s	PDF_RS	z,t,p,n		
PDF_R_g	PDF_RG	z,t,p,n		

C.5 LES averaged fields (LLES_RESOLVED=TRUE)

field	notation in the diac. file	dim.	if	comments
$<\tilde{u}^2>$	RES_U2	z,t,p		warning, contains both turbulent and gravity wave fields
$<\tilde{v}^2>$	RES_V2	z,t,p		
$<\tilde{w}^2>$	RES_W2	z,t,p		
$<\tilde{u}\,\tilde{v}>$	RES_UV	z,t,p		
$<\tilde{w}\,\tilde{u}>$	RES_WU	z,t,p		
$<\tilde{w}\,\tilde{v}>$	RES_WV	z,t,p		
$<\frac{1}{2}(\tilde{u}^2+\tilde{v}^2+\tilde{w}^2)>$	RES_KE	z,t,p		
$<\tilde{p}^2>$	RES_P2	z,t,p		
$<\tilde{u}\frac{\partial}{\partial z}\tilde{p}>$	RES_UPZ	z,t,p		
$<\tilde{v}\frac{\partial}{\partial z}\tilde{p}>$	RES_VPZ	z,t,p		
$<\tilde{w}\frac{\partial}{\partial z}\tilde{p}>$	RES_WPZ	z,t,p		
$<\tilde{\theta}\;\tilde{\theta_v}>$	RES_THTV	z,t,p	r_v	
$<\tilde{\theta_l}\;\tilde{\theta_v}>$	RES_TLTV	z,t,p	r_c	
$<\tilde{\theta}^2>$	RES_TH2	z,t,p		
$<\tilde{\theta_l}^2>$	RES_THL2	z,t,p	r_c	
$<\tilde{u}\tilde{\theta}>$	RES_UTH	z,t,p		
$<\tilde{v}\tilde{\theta}>$	RES_VTH	z,t,p		
$<\tilde{w}\tilde{\theta}>$	RES_WTH	z,t,p		
$<\tilde{u}\tilde{\theta_l}>$	RES_UTHL	z,t,p	r_c	
$<\tilde{v}\tilde{\theta_l}>$	RES_VTHL	z,t,p	r_c	
$<\tilde{w}\tilde{\theta_l}>$	RES_WTHL	z,t,p	r_c	
$<\tilde{u}\tilde{\theta_v}>$	RES_UTHV	z,t,p	r_v	
$<\tilde{v}\tilde{\theta_v}>$	RES_VTHV	z,t,p	r_v	
$<\tilde{w}\tilde{\theta_v}>$	RES_WTHV	z,t,p	r_v	

$< \tilde{r_v}^2 >$	RES_RV2	z,t,p	r_v	
$< \tilde{\theta}\tilde{r_v} >$	RES_THRV	z,t,p	r_v	
$< \tilde{\theta_l}\tilde{r_v} >$	RES_TLRV	z,t,p	r_c	
$< \tilde{\theta_v}\tilde{r_v} >$	RES_TVRV	z,t,p	r_v	
$< \tilde{u}\tilde{r_v} >$	RES_URV	z,t,p	r_v	
$< \tilde{v}\tilde{r_v} >$	RES_VRV	z,t,p	r_v	
$< \tilde{w}\tilde{r_v} >$	RES_WRV	z,t,p	r_v	
$< \tilde{r_c}^2 >$	RES_RC2	z,t,p	r_c	
$< \tilde{\theta}\tilde{r_c} >$	RES_THRC	z,t,p	r_c	
$< \tilde{\theta_l}\tilde{r_c} >$	RES_TLRC	z,t,p	r_c	
$< \tilde{\theta_v}\tilde{r_c} >$	RES_TVRC	z,t,p	r_c	
$< \tilde{u}\tilde{r_c} >$	RES_URC	z,t,p	r_c	warning, contains both turbulent and gravity wave fields
$< \tilde{v}\tilde{r_c} >$	RES_VRC	z,t,p	r_c	
$< \tilde{w}\tilde{r_c} >$	RES_WRC	z,t,p	r_c	
$< \tilde{r_i}^2 >$	RES_RI2	z,t,p	r_i	
$< \tilde{\theta}\tilde{r_i} >$	RES_THRI	z,t,p	r_i	
$< \tilde{\theta_l}\tilde{r_i} >$	RES_TLRI	z,t,p	r_i	
$< \tilde{\theta_v}\tilde{r_i} >$	RES_TVRI	z,t,p	r_i	
$< \tilde{u}\tilde{r_i} >$	RES_URI	z,t,p	r_i	
$< \tilde{v}\tilde{r_i} >$	RES_VRI	z,t,p	r_i	
$< \tilde{w}\tilde{r_i} >$	RES_WRI	z,t,p	r_i	
$< \tilde{w}\tilde{r_r} >$	RES_WRR	z,t,p	r_r	
Precipitation flux	INPRR3D	z,t,p	r_r	
Max Precipitation flux	MAXINPRR3D	z,t,p	r_r	
Evaporation flux	EVAP3D	z,t,p	r_r	
$< \tilde{s_v}^2 >$	RES_SV2	z,t,p,n	s_v	
$< \tilde{\theta}\tilde{s_v} >$	RES_THSV	z,t,p,n	s_v	
$< \tilde{\theta_l}\tilde{s_v} >$	RES_TLSV	z,t,p,n	s_v	
$< \tilde{\theta_v}\tilde{s_v} >$	RES_TVSV	z,t,p,n	s_v	
$< \tilde{u}\tilde{s_v} >$	RES_USV	z,t,p,n	s_v	
$< \tilde{v}\tilde{s_v} >$	RES_VSV	z,t,p,n	s_v	
$< \tilde{w}\tilde{s_v} >$	RES_WSV	z,t,p,n	s_v	
$< \tilde{w}^3 >$	RES_W3	z,t,p	s_v	
$< \tilde{w}\ \tilde{\theta_l}^2 >$	RES_WTL2	z,t,p	r_c	
$< \tilde{w}^2\ \tilde{\theta_l} >$	RES_W2TL	z,t,p	r_c	
$< \tilde{w}\ \tilde{r_t}^2 >$	RES_WRT2	z,t,p	r_c	
$< \tilde{w}^2\ \tilde{r_t} >$	RES_W2RT	z,t,p	r_c	
$< \tilde{w}\ \tilde{\theta_l}; \tilde{r_t} >$	RES_WTLRT	z,t,p	r_c	
$< \tilde{w}\ \tilde{r_v}^2 >$	RES_WRV2	z,t,p	r_v	
$< \tilde{w}^2\ \tilde{r_v} >$	RES_W2RV	z,t,p	r_v	
$< \tilde{w}\ \tilde{\theta_l}\tilde{r_v} >$	RES_WTLRV	z,t,p	r_v	if r_v and no r_c, replaced by $< \tilde{w}\ \tilde{\theta}\tilde{r_v} >$
$< \tilde{w}\ \tilde{r_c}^2 >$	RES_WRC2	z,t,p	r_c	
$< \tilde{w}^2\ \tilde{r_c} >$	RES_W2RC	z,t,p	r_c	
$< \tilde{w}\ \tilde{\theta_l}\tilde{r_c} >$	RES_WTLRC	z,t,p	r_c	
$< \tilde{w}\ \tilde{r_v}\tilde{r_c} >$	RE_WRVRC	z,t,p	r_c	

field	notation in diac. file	dim.	if	comments
$< \tilde{w}\ \tilde{r_i}^2 >$	RES_WRI2	z,t,p	r_i	
$< \tilde{w}^2\ \tilde{r_i} >$	RES_W2RI	z,t,p	r_i	
$< \tilde{w}\ \tilde{\theta_l}\tilde{r_i} >$	RE_WTLRI	z,t,p	r_i	
$< \tilde{w}\ \tilde{r_v}\tilde{r_i} >$	RE_WRVRI	z,t,p	r_i	
$< \tilde{w}\ \tilde{s_v}^2 >$	RES_WSV2	z,t,p,n	s_v	
$< \tilde{w}^2\ \tilde{s_v} >$	RES_W2SV	z,t,p,n	s_v	
$< \tilde{w}\ \tilde{\theta_l}\tilde{s_v} >$	RE_WTLSV	z,t,p,n	s_v	if no r_c, replaced by $< \tilde{w}\ \tilde{\theta}\tilde{s_v} >$
$< \tilde{w}\ \tilde{r_v}\tilde{s_v} >$	RE_WRVSV	z,t,p,n	r_v, s_v	
$< \tilde{\theta_l}\frac{\partial}{\partial z}\tilde{p} >$	RES_TLPZ	z,t,p		if no r_c, replaced by $< \tilde{\theta}\frac{\partial}{\partial z,t,p}\tilde{p} >$
$< \tilde{r_v}\frac{\partial}{\partial z}\tilde{p} >$	RES_RVPZ	z,t,p	r_v	
$< \tilde{r_c}\frac{\partial}{\partial z}\tilde{p} >$	RES_RCPZ	z,t,p	r_c	
$< \tilde{r_i}\frac{\partial}{\partial z}\tilde{p} >$	RES_RIPZ	z,t,p	r_i	
$< \tilde{s_v}\frac{\partial}{\partial z}\tilde{p} >$	RES_SVPZ	z,t,p,n	s_v	
$< \tilde{u}\ \boxed{\frac{1}{2}(\tilde{u}^2+\tilde{v}^2+\tilde{w}^2)}\ >$	RES_UKE	z,t,p		
$< \tilde{v}\ \boxed{\frac{1}{2}(\tilde{u}^2+\tilde{v}^2+\tilde{w}^2)}\ >$	RES_VKE	z,t,p		
$< \tilde{w}\ \boxed{\frac{1}{2}(\tilde{u}^2+\tilde{v}^2+\tilde{w}^2)}\ >$	RES_WKE	z,t,p		

C.6 LES averaged fields (LLES_SUBGRID=TRUE)

field	notation in diac. file	dim.	if	comments
$< \frac{1}{2}(\overline{u'^2}+\overline{v'^2}+\overline{w'^2}) >$	SBG_TKE	z,t,p		
$< \overline{u'^2} >$	SBG_U2	z,t,p		
$< \overline{v'^2} >$	SBG_V2	z,t,p		
$< \overline{w'^2} >$	SBG_W2	z,t,p		
$< \overline{u'v'} >$	SBG_UV	z,t,p		
$< \overline{w'u'} >$	SBG_WU	z,t,p		
$< \overline{w'v'} >$	SBG_WV	z,t,p		
$< \overline{\theta_l'^2} >$	SBG_THL2	z,t,p		if no r_c, replaced by $< \overline{\theta'^2} >$
$< \overline{u'\theta_l'} >$	SBG_UTHL	z,t,p		if no r_c, replaced by $< \overline{u'\theta'} >$
$< \overline{v'\theta_l'} >$	SBG_VTHL	z,t,p		if no r_c, replaced by $< \overline{v'\theta'} >$
$< \overline{w'\theta_l'} >$	SBG_WTHL	z,t,p		if no r_c, replaced by $< \overline{w'\theta'} >$

$< \overline{w'\theta'_v} >$	SBG_WTHV	z,t,p	r_v	
$< \overline{r'^2_t} >$	SBG_RT2	z,t,p	r_v	r_t is for total water
$< \overline{\theta'_l r'_t} >$	SBG_TLRT	z,t,p	r_v	if no r_c, replaced by $< \overline{\theta' r'_v} >$
$< \overline{u' r'_t} >$	SBG_URT	z,t,p	r_v	r_t is for total water
$< \overline{v' r'_t} >$	SBG_VRT	z,t,p	r_v	r_t is for total water
$< \overline{w' r'_t} >$	SBG_WRT	z,t,p	r_v	r_t is for total water
$< \overline{r'^2_c} >$	SBG_RC2	z,t,p	r_c	
$< \overline{u' r'_c} >$	SBG_URC	z,t,p	r_c	
$< \overline{v' r'_c} >$	SBG_VRC	z,t,p	r_c	
$< \overline{w' r'_c} >$	SBG_WRC	z,t,p	r_c	
$< \overline{u' s'_v} >$	SBG_USV	z,t,p,n	s_v	
$< \overline{v' s'_v} >$	SBG_VSV	z,t,p,n	s_v	
$< \overline{w' s'_v} >$	SBG_WSV	z,t,p,n	s_v	
$< \overline{u'} \left[\frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}) \right] >$	SBG_UTKE	z,t,p		
$< \overline{v'} \left[\frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}) \right] >$	SBG_VTKE	z,t,p		
$< \overline{w'} \left[\frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}) \right] >$	SBG_WTKE	z,t,p		
$< \overline{\theta'_{lUpdraft}} >$	SBG_THLUP	z,t,p		
$< \overline{r'_{tUpdraft}} >$	SBG_RTUP	z,t,p	r_v	
$< \overline{r'_{vUpdraft}} >$	SBG_RVUP	z,t,p	r_v	
$< \overline{r'_{cUpdraft}} >$	SBG_RCUP	z,t,p	r_c	
$< \overline{r'_{iUpdraft}} >$	SBG_RIUP	z,t,p	r_i	
$< \overline{\theta}_{lUpdraftMassFlux} >$	THLUP_MF	z,t,p	r_i	
$< \overline{r}_{tUpdraftMassFlux} >$	RTUP_MF	z,t,p	r_v	
$< \overline{r}_{vUpdraftMassFlux} >$	RVUP_MF	z,t,p	r_v	
$< \overline{r}_{cUpdraftMassFlux} >$	RCUP_MF	z,t,p	r_c	
$< \overline{r}_{iUpdraftMassFlux} >$	RIUP_MF	z,t,p	r_i	
$< \overline{W}_{UpdraftMassFlux} >$	WUP_MF	z,t,p		
$< \overline{MassFlux}_{UpdraftMassFlux} >$	MAFLX_MF	z,t,p		
$< \overline{Detr}_{UpdraftMassFlux} >$	DETR_MF	z,t,p		
$< \overline{Entr}_{UpdraftMassFlux} >$	ENTR_MF	z,t,p		
$< \overline{Frac}_{UpdraftMassFlux} >$	FRCUP_MF	z,t,p		
$< \overline{\theta}_{vUpdraftMassFlux} >$	THVUP_MF	z,t,p		
$< \overline{w'\theta'}_{lMassFlux} >$	WTHL_MF	z,t,p		
$< \overline{w' r'}_{tMassFlux} >$	WRT_MF	z,t,p		
$< \overline{w'\theta'}_{vMassFlux} >$	WTHV_MF	z,t,p		
$< \overline{w'u'}_{MassFlux} >$	WU_MF	z,t,p		
$< \overline{w'v'}_{MassFlux} >$	WV_MF	z,t,p		

C.7 LES averaged fields (LLES_UPDRAFT=TRUE)

$< f_{up} >$	UP_FRAC	z,t		updraft fraction
$< w >_{up}$	UP_W	z,t		from now, computations are made only on the cartesian mask
$< \theta >_{up}$	UP_TH	z,t		
$< \theta_l >_{up}$	UP_THL	z,t	r_c	
$< \theta_v >_{up}$	UP_THV	z,t	r_v	
$< \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2) >_{up}$	UP_KE	z,t		
$< \frac{1}{2}(\tilde{u'^2} + \tilde{v'^2} + \tilde{w'^2}) >_{up}$	UP_TKE	z,t		
$< r_v >_{up}$	UP_RV	z,t	r_v	
$< r_c >_{up}$	UP_RC	z,t	r_c	
$< r_r >_{up}$	UP_RR	z,t	r_r	
$< r_i >_{up}$	UP_RI	z,t	r_i	
$< r_s >_{up}$	UP_RS	z,t	r_s	
$< r_g >_{up}$	UP_RG	z,t	r_g	
$< r_h >_{up}$	UP_RH	z,t	r_h	
$< s_v >_{up}$	UP_SV	z,t,n	s_v	
$< \tilde{\theta}^2 >_{up}$	UP_TH2	z,t		
$< \tilde{\theta_l}^2 >_{up}$	UP_THL2	z,t	r_c	
$< \tilde{\theta}\ \tilde{\theta_v} >_{up}$	UP_THTV	z,t	r_v	
$< \tilde{\theta_l}\ \tilde{\theta_v} >_{up}$	UP_TLTV	z,t	r_c	
$< \tilde{w}\ \tilde{\theta} >_{up}$	UP_WTH	z,t		
$< \tilde{w}\ \tilde{\theta_l} >_{up}$	UP_WTHL	z,t	r_c	
$< \tilde{w}\ \tilde{\theta_v} >_{up}$	UP_WTHV	z,t	r_v	
$< \tilde{r_v}^2 >_{up}$	UP_RV2	z,t	r_v	
$< \tilde{\theta}\ \tilde{r_v} >_{up}$	UP_THRV	z,t	r_v	
$< \tilde{\theta_l}\ \tilde{r_v} >_{up}$	UP_TLRV	z,t	r_c	
$< \tilde{\theta_v}\ \tilde{r_v} >_{up}$	UP_TVRV	z,t	r_v	
$< \tilde{w}\ \tilde{r_v} >_{up}$	UP_WRV	z,t	r_v	
$< \tilde{r_c}^2 >_{up}$	UP_RC2	z,t	r_c	
$< \tilde{\theta}\ \tilde{r_c} >_{up}$	UP_THRC	z,t	r_c	
$< \tilde{\theta_l}\ \tilde{r_c} >_{up}$	UP_TLRC	z,t	r_c	
$< \tilde{\theta_v}\ \tilde{r_c} >_{up}$	UP_TVRC	z,t	r_c	
$< \tilde{w}\ \tilde{r_c} >_{up}$	UP_WRC	z,t	r_c	
$< \tilde{r_i}^2 >_{up}$	UP_RI2	z,t	r_i	
$< \tilde{\theta}\ \tilde{r_i} >_{up}$	UP_THRI	z,t	r_i	
$< \tilde{\theta_l}\ \tilde{r_i} >_{up}$	UP_TLRI	z,t	r_i	
$< \tilde{\theta_v}\ \tilde{r_i} >_{up}$	UP_TVRI	z,t	r_i	
$< \tilde{w}\ \tilde{r_i} >_{up}$	UP_WRI	z,t	r_i	
$< \tilde{s_v}^2 >_{up}$	UP_SV2	z,t,n	s_v	
$< \tilde{\theta}\ \tilde{s_v} >_{up}$	UP_THSV	z,t,n		
$< \tilde{\theta_l}\ \tilde{s_v} >_{up}$	UP_TLSV	z,t,n	r_c, s_v	
$< \tilde{\theta_v}\ \tilde{s_v} >_{up}$	UP_TVSV	z,t,n	r_v, s_v	
$< \tilde{w}\ \tilde{s_v} >_{up}$	UP_WSV	z,t,n	s_v	

C.8 LES averaged fields (LLES_DOWNDRAFT=TRUE)

$< f_{dw} >$	DW_FRAC	z,t		downdraft fraction
$< w >_{dw}$	DW_W	z,t		
$< \theta >_{dw}$	DW_TH	z,t		
$< \theta_l >_{dw}$	DW_THL	z,t	r_c	
$< \theta_v >_{dw}$	DW_THV	z,t	r_v	
$< \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2) >_{dw}$	DW_KE	z,t		
$< \frac{1}{2}(\tilde{u'^2} + \tilde{v'^2} + \tilde{w'^2}) >_{dw}$	DW_TKE	z,t		
$< r_v >_{dw}$	DW_RV	z,t	r_v	
$< r_c >_{dw}$	DW_RC	z,t	r_c	
$< r_r >_{dw}$	DW_RR	z,t	r_r	
$< r_i >_{dw}$	DW_RI	z,t	r_i	
$< r_s >_{dw}$	DW_RS	z,t	r_s	
$< r_g >_{dw}$	DW_RG	z,t	r_g	
$< r_h >_{dw}$	DW_RH	z,t	r_h	
$< s_v >_{dw}$	DW_SV	z,t,n	s_v	
$< \tilde{\theta}^2 >_{dw}$	DW_TH2	z,t		
$< \tilde{\theta_l}^2 >_{dw}$	DW_THL2	z,t	r_c	
$< \tilde{\theta}\ \tilde{\theta_v} >_{dw}$	DW_THTV	z,t	r_v	
$< \tilde{\theta_l}\ \tilde{\theta_v} >_{dw}$	DW_TLTV	z,t	r_c	
$< \tilde{w}\ \tilde{\theta} >_{dw}$	DW_WTH	z,t		
$< \tilde{w}\ \tilde{\theta_l} >_{dw}$	DW_WTHL	z,t	r_c	
$< \tilde{w}\ \tilde{\theta_v} >_{dw}$	DW_WTHV	z,t	r_v	
$< \tilde{r_v}^2 >_{dw}$	DW_RV2	z,t	r_v	
$< \tilde{\theta}\ \tilde{r_v} >_{dw}$	DW_THRV	z,t	r_v	
$< \tilde{\theta_l}\ \tilde{r_v} >_{dw}$	DW_TLRV	z,t	r_c	
$< \tilde{\theta_v}\ \tilde{r_v} >_{dw}$	DW_TVRV	z,t	r_v	
$< \tilde{w}\ \tilde{r_v} >_{dw}$	DW_WRV	z,t	r_v	
$< \tilde{r_c}^2 >_{dw}$	DW_RC2	z,t	r_c	
$< \tilde{\theta}\ \tilde{r_c} >_{dw}$	DW_THRC	z,t	r_c	
$< \tilde{\theta_l}\ \tilde{r_c} >_{dw}$	DW_TLRC	z,t	r_c	
$< \tilde{\theta_v}\ \tilde{r_c} >_{dw}$	DW_TVRC	z,t	r_c	
$< \tilde{w}\ \tilde{r_c} >_{dw}$	DW_WRC	z,t	r_c	
$< \tilde{r_i}^2 >_{dw}$	DW_RI2	z,t	r_i	
$< \tilde{\theta}\ \tilde{r_i} >_{dw}$	DW_THRI	z,t	r_i	
$< \tilde{\theta_l}\ \tilde{r_i} >_{dw}$	DW_TLRI	z,t	r_i	
$< \tilde{\theta_v}\ \tilde{r_i} >_{dw}$	DW_TVRI	z,t	r_i	
$< \tilde{w}\ \tilde{r_i} >_{dw}$	DW_WRI	z,t	r_i	
$< \tilde{s_v}^2 >_{dw}$	DW_SV2	z,t,n	s_v	
$< \tilde{\theta}\ \tilde{s_v} >_{dw}$	DW_THSV	z,t,n		
$< \tilde{\theta_l}\ \tilde{s_v} >_{dw}$	DW_TLSV	z,t,n	r_c, s_v	
$< \tilde{\theta_v}\ \tilde{s_v} >_{dw}$	DW_TVSV	z,t,n	r_v, s_v	
$< \tilde{w}\ \tilde{s_v} >_{dw}$	DW_WSV	z,t,n	s_v	

C.9 LES averaged surface fields

field	notation in diac. file	dimen-sion	if	comments
$<\overline{w'\theta'}_{surf}>$	Q0	t		surface sensible flux
$<\overline{w'r'_v}_{surf}>$	E0	t	r_v	surface latent flux
$<\overline{w's'_v}_{surf}>$	E0	t,n	s_v	surface scalar flux
$u_* = \left\{<\overline{u'w'}_{surf}>^2+<\overline{v'w'}_{surf}>^2\right\}^{\frac{1}{4}}$	U*	t		friction velocity
$w_* = \left\{<\frac{g}{\theta}><\overline{w'\theta'_v}_{surf}><h>\right\}^{\frac{1}{3}}$	W*	t		convective velocity if positive surface buoyancy flux
$<h>$	BL_H	t		boundary layer height
$<L_{MO}>$	L_MO	t		Monin-Obukhov length
$\int TKEdz$	INT_TKE	t		vertical integrated TKE
	ZCB	t		cloud base height
CF	ZCFTOT	t	r_c	total cloud cover
$\int \rho(r_c+r_r)$	LWP	t	r_c	Cloud water path
VAR_LWP	LWPVAR	t	r_c	LWP variance
$\int \rho r_r$	RWP	t	r_r	Rain water path
$INPRR$	INST_PREC	t	r_r	Inst. precip. rate
	RAIN_PREC	t	r_r	INPRR over rainy grids
$ACPRR$	ACCU_PREC	t	r_r	Accum. precip. rate
H_CFmax	ZMAXCF	t	r_c	Height of cloud fraction maximum

C.10 Other LES averaged fields

field	notation in diac. file	dimen-sion	if	comments
sw_{up}	SWU	z,t,p		SW upward radiative flux
sw_{down}	SWD	z,t,p		SW downward radiative flux
lw_{up}	LWU	z,t,p		LW upward radiative flux
lw_{down}	LWD	z,t,p		LW downward radiative flux
$dthrad_{sw}$	DTHRADSW	z,t,p		SW radiative temperature tendency
$dthrad_{lw}$	DTHRADLW	z,t,p		LW radiative temperature tendency
Mean Effective Radius	RADEFF	z,t,p		Mean effective radius

C.11 LES 2 points correlations

field	notation in the diac. file	dim.	if	comments
$< \tilde{u}(x,y)\ \tilde{u}(x+l_x,y) >$	CI_UU	l_x,2,z,t		
$< \tilde{u}(x,y)\ \tilde{u}(x,y+l_y) >$	CJ_UU	l_y,2,z,t		
$< \tilde{v}(x,y)\ \tilde{v}(x+l_x,y) >$	CI_VV	l_x,2,z,t		
$< \tilde{v}(x,y)\ \tilde{v}(x,y+l_y) >$	CJ_VV	l_y,2,z,t		
$< \tilde{w}(x,y)\ \tilde{w}(x+l_x,y) >$	CI_WW	l_x,2,z,t		
$< \tilde{w}(x,y)\ \tilde{w}(x,y+l_y) >$	CJ_WW	l_y,2,z,t		
$< \tilde{u}(x,y)\ \tilde{v}(x+l_x,y) >$	CI_UV	l_x,2,z,t		
$< \tilde{u}(x,y)\ \tilde{v}(x,y+l_y) >$	CJ_UV	l_y,2,z,t		
$< \tilde{w}(x,y)\ \tilde{u}(x+l_x,y) >$	CI_WU	l_x,2,z,t		
$< \tilde{w}(x,y)\ \tilde{u}(x,y+l_y) >$	CJ_WV	l_y,2,z,t		
$< \tilde{\theta}(x,y)\ \tilde{\theta}(x+l_x,y) >$	CI_THTH	l_x,2,z,t		
$< \tilde{\theta}(x,y)\ \tilde{\theta}(x,y+l_y) >$	CJ_THTH	l_y,2,z,t		
$< \tilde{\theta}_l(x,y)\ \tilde{\theta}_l(x+l_x,y) >$	CI_TLTL	l_x,2,z,t	r_c	
$< \tilde{\theta}_l(x,y)\ \tilde{\theta}_l(x,y+l_y) >$	CJ_TLTL	l_y,2,z,t	r_c	
$< \tilde{w}(x,y)\ \tilde{\theta}(x+l_x,y) >$	CI_WTH	l_x,2,z,t		
$< \tilde{w}(x,y)\ \tilde{\theta}(x,y+l_y) >$	CJ_WTH	l_y,2,z,t		
$< \tilde{w}(x,y)\ \tilde{\theta}_l(x+l_x,y) >$	CI_WTHL	l_x,2,z,t	r_c	
$< \tilde{w}(x,y)\ \tilde{\theta}_l(x,y+l_y) >$	CJ_WTHL	l_y,2,z,t	r_c	
$< \tilde{r_v}(x,y)\ \tilde{r_v}(x+l_x,y) >$	CI_RVRV	l_x,2,z,t	r_v	
$< \tilde{r_v}(x,y)\ \tilde{r_v}(x,y+l_y) >$	CJ_RVRV	l_y,2,z,t	r_v	
$< \tilde{\theta}(x,y)\ \tilde{r_v}(x+l_x,y) >$	CI_THRV	l_x,2,z,t	r_v	
$< \tilde{\theta}(x,y)\ \tilde{r_v}(x,y+l_y) >$	CJ_THRV	l_y,2,z,t	r_v	
$< \tilde{\theta}_l(x,y)\ \tilde{r_v}(x+l_x,y) >$	CI_TLRV	l_x,2,z,t	r_c	
$< \tilde{\theta}_l(x,y)\ \tilde{r_v}(x,y+l_y) >$	CJ_TLRV	l_y,2,z,t	r_c	
$< \tilde{w}(x,y)\ \tilde{r_v}(x+l_x,y) >$	CI_WRV	l_x,2,z,t	r_v	
$< \tilde{w}(x,y)\ \tilde{r_v}(x,y+l_y) >$	CJ_WRV	l_y,2,z,t	r_v	
$< \tilde{r_c}(x,y)\ \tilde{r_c}(x+l_x,y) >$	CI_RCRC	l_x,2,z,t	r_c	
$< \tilde{r_c}(x,y)\ \tilde{r_c}(x,y+l_y) >$	CJ_RCRC	l_y,2,z,t	r_c	
$< \tilde{\theta}(x,y)\ \tilde{r_c}(x+l_x,y) >$	CI_THRC	l_x,2,z,t	r_c	
$< \tilde{\theta}(x,y)\ \tilde{r_c}(x,y+l_y) >$	CJ_THRC	l_y,2,z,t	r_c	
$< \tilde{\theta}_l(x,y)\ \tilde{r_c}(x+l_x,y) >$	CI_TLRC	l_x,2,z,t	r_c	
$< \tilde{\theta}_l(x,y)\ \tilde{r_c}(x,y+l_y) >$	CJ_TLRC	l_y,2,z,t	r_c	
$< \tilde{w}(x,y)\ \tilde{r_c}(x+l_x,y) >$	CI_WRC	l_x,2,z,t	r_c	
$< \tilde{w}(x,y)\ \tilde{r_c}(x,y+l_y) >$	CJ_WRC	l_y,2,z,t	r_c	
$< \tilde{r_i}(x,y)\ \tilde{r_i}(x+l_x,y) >$	CI_RIRI	l_x,2,z,t	r_i	
$< \tilde{r_i}(x,y)\ \tilde{r_i}(x,y+l_y) >$	CJ_RIRI	l_y,2,z,t	r_i	
$< \tilde{\theta}(x,y)\ \tilde{r_i}(x+l_x,y) >$	CI_THRI	l_x,2,z,t	r_i	
$< \tilde{\theta}(x,y)\ \tilde{r_i}(x,y+l_y) >$	CJ_THRI	l_y,2,z,t	r_i	
$< \tilde{\theta}_l(x,y)\ \tilde{r_i}(x+l_x,y) >$	CI_TLRI	l_x,2,z,t	r_i	
$< \tilde{\theta}_l(x,y)\ \tilde{r_i}(x,y+l_y) >$	CJ_TLRI	l_y,2,z,t	r_i	

field	notation in the diac. file	dim.	if	comments
$< \tilde{w}(x,y)\ \tilde{r_i}(x+l_x,y) >$	CI_WRI	l_x,2,z,t	r_i	
$< \tilde{w}(x,y)\ \tilde{r_i}(x,y+l_y) >$	CJ_WRI	l_y,2,z,t	r_i	
$< \tilde{s_v}(x,y)\ \tilde{s_v}(x+l_x,y) >$	CI_SVSV	l_x,2,z,t,n	s_v	
$< \tilde{s_v}(x,y)\ \tilde{s_v}(x,y+l_y) >$	CJ_SVSV	l_y,2,z,t,n	s_v	
$< \tilde{w}(x,y)\ \tilde{s_v}(x+l_x,y) >$	CI_WSV	l_x,2,z,t,n	s_v	
$< \tilde{w}(x,y)\ \tilde{s_v}(x,y+l_y) >$	CJ_WSV	l_y,2,z,t,n	s_v	

C.12 LES spectra

field	notation in the diac. file	dim.	if	comments
$\frac{1}{2\pi L}\int_L < \tilde{u}(x,y)\ \tilde{u}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_UU	k_x,2,z,t		
$\frac{1}{2\pi L}\int_L < \tilde{u}(x,y)\ \tilde{u}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_UU	k_y,2,z,t		
$\frac{1}{2\pi L}\int_L < \tilde{v}(x,y)\ \tilde{v}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_VV	k_x,2,z,t		
$\frac{1}{2\pi L}\int_L < \tilde{v}(x,y)\ \tilde{v}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_VV	k_y,2,z,t		
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{w}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_WW	k_x,2,z,t		
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{w}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_WW	k_y,2,z,t		
$\frac{1}{2\pi L}\int_L < \tilde{u}(x,y)\ \tilde{v}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_UV	k_x,2,z,t		
$\frac{1}{2\pi L}\int_L < \tilde{u}(x,y)\ \tilde{v}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_UV	k_y,2,z,t		dimension 2 is for real and imaginary parts
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{u}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_WU	k_x,2,z,t		
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{u}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_WV	k_y,2,z,t		
$\frac{1}{2\pi L}\int_L < \tilde{\theta}(x,y)\ \tilde{\theta}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_THTH	k_x,2,z,t		
$\frac{1}{2\pi L}\int_L < \tilde{\theta}(x,y)\ \tilde{\theta}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_THTH	k_y,2,z,t		
$\frac{1}{2\pi L}\int_L < \tilde{\theta_l}(x,y)\ \tilde{\theta_l}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_TLTL	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{\theta_l}(x,y)\ \tilde{\theta_l}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_TLTL	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{\theta}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_WTH	k_x,2,z,t		

field	notation in the diac. file	dim.	if	comments
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{\theta}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_WTH	k_y,2,z,t		dimension 2 is for real and imaginary parts
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{\theta_l}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_WTHL	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{\theta_l}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_WTHL	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{r_v}(x,y)\ \tilde{r_v}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_RVRV	k_x,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L < \tilde{r_v}(x,y)\ \tilde{r_v}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_RVRV	k_y,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L < \tilde{\theta}(x,y)\ \tilde{r_v}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_THRV	k_x,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L < \tilde{\theta}(x,y)\ \tilde{r_v}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_THRV	k_y,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L < \tilde{\theta_l}(x,y)\ \tilde{r_v}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_TLRV	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{\theta_l}(x,y)\ \tilde{r_v}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_TLRV	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{r_v}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_WRV	k_x,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{r_v}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_WRV	k_y,2,z,t	r_v	
$\frac{1}{2\pi L}\int_L < \tilde{r_c}(x,y)\ \tilde{r_c}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_RCRC	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{r_c}(x,y)\ \tilde{r_c}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_RCRC	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{\theta}(x,y)\ \tilde{r_c}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_THRC	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{\theta}(x,y)\ \tilde{r_c}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_THRC	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{\theta_l}(x,y)\ \tilde{r_c}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_TLRC	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{\theta_l}(x,y)\ \tilde{r_c}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_TLRC	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{r_c}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_WRC	k_x,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{r_c}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_WRC	k_y,2,z,t	r_c	
$\frac{1}{2\pi L}\int_L < \tilde{r_i}(x,y)\ \tilde{r_i}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_RIRI	k_x,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L < \tilde{r_i}(x,y)\ \tilde{r_i}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_RIRI	k_y,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L < \tilde{\theta}(x,y)\ \tilde{r_i}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_THRI	k_x,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L < \tilde{\theta}(x,y)\ \tilde{r_i}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_THRI	k_y,2,z,t	r_i	

field	notation in the diac. file	dim.	if	comments
$\frac{1}{2\pi L}\int_L < \tilde{\theta}_l(x,y)\ \tilde{r}_i(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_TLRI	k_x,2,z,t	r_i	dimension 2 is for real and imaginary parts
$\frac{1}{2\pi L}\int_L < \tilde{\theta}_l(x,y)\ \tilde{r}_i(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_TLRI	k_y,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{r}_i(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_WRI	k_x,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{r}_i(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_WRI	k_y,2,z,t	r_i	
$\frac{1}{2\pi L}\int_L < \tilde{s_v}(x,y)\ \tilde{s_v}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_SVSV	k_x,2,z,t,n	s_v	
$\frac{1}{2\pi L}\int_L < \tilde{s_v}(x,y)\ \tilde{s_v}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_SVSV	k_y,2,z,t,n	s_v	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{s_v}(x+l_x,y) > e^{-ik_xl_x}dl_x$	SI_WSV	k_x,2,z,t,n	s_v	
$\frac{1}{2\pi L}\int_L < \tilde{w}(x,y)\ \tilde{s_v}(x,y+l_y) > e^{-ik_yl_y}dl_y$	SJ_WSV	k_y,2,z,t,n	s_v	

C.13 Budget of (resolved + subgrid) turbulent quantities

C.13.1 Budget of total turbulent kinetic energy

All terms of the equation of $\frac{\partial}{\partial t}(E+e)$ are computed and stored in the diachronic group BU_KE. Here, e and E denote the subgrid and resolved Tke respectively:

$$< e > = < \frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}) > \qquad\qquad < E > = < \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2) >$$

Here are **main terms** of the equations for the horizontal mean of subgrid Tke and resolved Tke, in the frame of Boussinesq approximation. Note that the computations of the budgets terms are done with the complete equation set and discretization of MESONH. The equations here are simplified only for the sake of easier understanding. Other terms can arise from the parametrizations of MESONH, and will also be taken into account in the budget.

$$\frac{\partial}{\partial t} < e > = \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < e >}^{ADVM} \overbrace{- < \tilde{u_\alpha} \frac{\partial}{\partial x_\alpha} e >}^{ADVR} \overbrace{- \frac{1}{< \rho >} < u'_\alpha \overline{\frac{\partial p'}{\partial x_\alpha}} >}^{PRES} \overbrace{- < \frac{\partial}{\partial x_\alpha} \overline{u'_\alpha e} >}^{TR}$$
$$\underbrace{- < \overline{u'_\alpha u'_\beta} > \frac{\partial < u_\beta >}{\partial x_\alpha}}_{DPM} \underbrace{- < \overline{u'_\alpha u'_\beta} \frac{\partial \tilde{u_\beta}}{\partial x_\alpha} >}_{DPR} + \underbrace{< \beta \overline{w'\theta'_v} >}_{TP} \underbrace{- < \epsilon >}_{DISS}$$

$$\frac{\partial}{\partial t} < E > = \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < E >}^{ADV} \overbrace{- \frac{1}{< \rho >} < \tilde{u_\alpha} \frac{\partial \tilde{p}}{\partial x_\alpha} >}^{PRES} \overbrace{- < \tilde{u_\alpha} \tilde{u_\beta} > \frac{\partial < u_\beta >}{\partial x_\alpha}}^{DP}$$
$$+ \underbrace{< \beta \tilde{w} \tilde{\theta_v} >}_{TP} \underbrace{- \frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha} E >}_{TR} \underbrace{- < \tilde{u_\alpha} \frac{\partial}{\partial x_\beta} \overline{u'_\alpha u'_\beta} >}_{SBGT} + ...$$

The terms of (spectral) transport from resolved to subgrid motions is SBTG in the equation of $< E >$ (sink), and ADVR and DPR in the equation of $< e >$ (sources). One should note that:

$$\mathrm{ADVR} + \mathrm{DPR} = -\mathrm{SGBT}$$

<u>Note in case of gridnesting</u>

In case of 2way gridnesting, the subgrid scheme is not alone to influence the resolved motions due to subgrid scale. Part of the job is done by the averaged of the smaller-scale models. The terms of (spectral) transport from resolved to subgrid motions are then both SBTG and NEST in the equation of $< E >$ (sinks). Therefore

$$\mathrm{ADVR} + \mathrm{D\hat{P}R} = -(\mathrm{SGBT} + \mathrm{NEST})$$

Where $\hat{\mathrm{DPR}}$ is the dynamical production that should produce the subgrid-scale model to equilibrate the sink at resolved scale.

field	notation in diac. file	processus name	dim.	comments
$-\frac{\partial}{\partial t} < e >$	BU_KE	SBG TEND	z,t	(opposite of) tendency of $< e >$
$- < \overline{u'w'} > \frac{\partial}{\partial z} < u >$ $- < \overline{v'w'} > \frac{\partial}{\partial z} < v >$	BU_KE	SBG DP M	z,t	dyn. prod. by mean gradients
$< -\overline{u'_\alpha u'_\beta} \frac{\partial}{\partial x_\beta} \tilde{u_\alpha} >$	BU_KE	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$- < w > \frac{\partial}{\partial z} < e >$	BU_KE	SBG ADVM	z,t	advection by mean flow
$- < \tilde{u_\alpha} \frac{\partial}{\partial x_\alpha} e >$	BU_KE	SBG ADVR	z,t	advection by resolved flow
$-W_{forc} \frac{\partial}{\partial z} < e >$	BU_KE	SBG FORC	z,t	advection by large-scale W forcing
$- < \frac{\partial}{\partial z} \overline{w'e} >$	BU_KE	SBG TR	z,t	subgrid turbulent transport
$-\frac{1}{<\rho>} < \overline{u'_\alpha \frac{\partial p'}{\partial x_\alpha}} >$	BU_KE	SBG PRES	z,t	subgrid pressure-correlation term
$< \beta \overline{w'\theta'_v} >$	BU_KE	SBG TP	z,t	thermal production
$- < \epsilon >$	BU_KE	SBG DISS	z,t	dissipation
numerical diffusion of $< e >$	BU_KE	SBG NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $< e >$	BU_KE	SBG RELA	z,t	sponge layer relaxation
miscellaneous	BU_KE	SBG MISC	z,t	...
residual of budget of $< e >$	BU_KE	SBG RESI	z,t	must be zero
$-\frac{\partial}{\partial t} < E >$	BU_KE	RES TEND	z,t	(opposite of) tendency of $< E >$
$- < w > \frac{\partial}{\partial z} < E >$	BU_KE	RES ADV	z,t	advection by mean flow
$-W_{forc} \frac{\partial}{\partial z} < E >$	BU_KE	RES FORC	z,t	advection by large-scale W forcing
$- < \tilde{u}\tilde{w} > \frac{\partial}{\partial z} < u >$ $- < \tilde{v}\tilde{w} > \frac{\partial}{\partial z} < v >$	BU_KE	RES DP	z,t	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha} E >$	BU_KE	RES TR	z,t	transport of resolved Tke by itself
$-\frac{1}{<\rho>} < \tilde{u_\alpha} \frac{\partial \tilde{p}}{\partial x_\alpha} >$	BU_KE	RES PRES	z,t	pressure-correlations
$< \beta \tilde{w} \tilde{\theta_v} >$	BU_KE	RES TP	z,t	thermal production
$- < \tilde{u_\alpha} \frac{\partial}{\partial x_\beta} \overline{u'_\alpha u'_\beta} >$	BU_KE	RES SBGT	z,t	sink due to subgrid turbulence

field	notation in diac. file	processus name	dim.	comments
Coriolis terms	BU_KE	RES CORI	z,t	should be zero for $< E >$
numerical diffusion of $< E >$	BU_KE	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $< E >$	BU_KE	RES RELA	z,t	sponge layer relaxation
2way nesting of $< E >$	BU_KE	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_KE	RES MISC	z,t	curvature terms, ...
residual of budget of $< E >$	BU_KE	RES RESI	z,t	must be zero

Note that if a term is zero, because the process accounting for it is not activated in the model, the term is not listed in the diachronic file. So, in order to know which terms have been computed and stored, use the command 'print BU_KE proc' in diaprog.

C.13.2 Budget of total (liquid) temperature flux

All terms of the equation of $\frac{\partial}{\partial t}(< \tilde{w}\tilde{\theta}_l > + < \overline{w'\theta'_l} >)$ are computed and stored in the diachronic group BU_WTHL. All comments made for the total Tke equation are valid here.

$$
\begin{aligned}
\frac{\partial}{\partial t} < \overline{w'\theta'_l} >= \; & \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < \overline{w'\theta'_l} >}^{ADVM} \; \overbrace{- < \tilde{u_\alpha} \frac{\partial}{\partial x_\alpha} \overline{w'\theta'_l} >}^{ADVR} \quad \overbrace{- < \overline{u'_\alpha w'} > \frac{\partial < \theta_l >}{\partial x_\alpha} - < \overline{u'_\alpha \theta'_l} > \frac{\partial < w >}{\partial x_\alpha}}^{DPM} \\
& \underbrace{- < \overline{u'_\alpha w'} \frac{\partial \tilde{\theta}_l}{\partial x_\alpha} > - < \overline{u'_\alpha \theta'_l} \frac{\partial \tilde{w}}{\partial x_\alpha} >}_{DPR} \; + \underbrace{< \beta \overline{\theta'_l \theta'_v} >}_{TP} \quad \underbrace{- \frac{1}{< \rho >} < \overline{\theta'_l \frac{\partial p'}{\partial z}} >}_{PRES} \quad \underbrace{- < \frac{\partial}{\partial x_\alpha} \overline{u'_\alpha w' \theta'_l} >}_{TR}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial}{\partial t} < \tilde{w}\tilde{\theta}_l >= \; & \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < \tilde{w}\tilde{\theta}_l >}^{ADV} \; \overbrace{- \frac{1}{< \rho >} < \tilde{\theta}_l \frac{\partial \tilde{p}}{\partial x_\alpha} >}^{PRES} \quad \overbrace{- < \tilde{u_\alpha} \tilde{w} > \frac{\partial < \theta_l >}{\partial x_\alpha} - < \tilde{u_\alpha} \tilde{\theta}_l > \frac{\partial < w >}{\partial x_\alpha}}^{DP} \\
& + \underbrace{< \beta \tilde{\theta}_l \tilde{\theta}_v >}_{TP} \quad \underbrace{- \frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha} \tilde{w} \tilde{\theta}_l >}_{TR} \quad \underbrace{- < \tilde{w} \frac{\partial}{\partial x_\alpha} \overline{u'_\alpha \theta'_l} > - < \tilde{\theta}_l \frac{\partial}{\partial x_\alpha} \overline{u'_\alpha w'} >}_{SBGT} \; + ...
\end{aligned}
$$

field	notation in diac. file	processus name	dim.	comments
$-<\overline{w'^2}>\frac{\partial}{\partial z}<\theta_l>$	BU_WTHL	SBG DP M	z,t	dyn. prod. by mean gradient
$<-\overline{w'^2}\frac{\partial}{\partial z}\tilde{\theta}_l>$	BU_WTHL	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$-<\frac{\partial}{\partial z}\overline{w'^2\theta'_l}>$	BU_WTHL	SBG TR	z,t	subgrid turbulent transport
$-\frac{1}{<\rho>}<\overline{\theta'_l\frac{\partial p'}{\partial z}}>$	BU_WTHL	SBG PRES	z,t	subgrid pressure-correlation term
$<\beta\overline{\theta'_l\theta'_v}>$	BU_WTHL	SBG TP	z,t	thermal production
residual of budget of $<\overline{w'\theta'_l}>$	BU_WTHL	SBG RESI	z,t	must be small
$-\frac{\partial}{\partial t}<\tilde{w}\tilde{\theta}_l>$	BU_WTHL	RES TEND	z,t	(opposite of) tendency of $<\tilde{w}\tilde{\theta}_l>$
$-<w>\frac{\partial}{\partial z}<\tilde{w}\tilde{\theta}_l>$	BU_WTHL	RES ADV	z,t	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z}<\tilde{w}\tilde{\theta}_l>$	BU_WTHL	RES FORC	z,t	advection by large-scale W forcing
$-<\tilde{w}^2>\frac{\partial}{\partial z}<\theta_l>$ $-<\tilde{w}\tilde{\theta}_l>\frac{\partial}{\partial z}<w>$	BU_WTHL	RES DP	z,t	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{w}\tilde{\theta}_l>$	BU_WTHL	RES TR	z,t	transport of resolved flux by itself
$-\frac{1}{<\rho>}<\tilde{\theta}_l\frac{\partial\tilde{p}}{\partial z}>$	BU_WTHL	RES PRES	z,t	pressure-correlations
$<\beta\tilde{\theta}_l\tilde{\theta}_v>$	BU_WTHL	RES TP	z,t	thermal production
$-<\tilde{w}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha\theta'_l}>$ $-<\tilde{\theta}_l\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'}>$	BU_WTHL	RES SBGT	z,t	sink due to subgrid turbulence
Coriolis terms	BU_WTHL	RES CORI	z,t	
numerical diffusion of $<\tilde{w}\tilde{\theta}_l>$	BU_WTHL	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $<\tilde{w}\tilde{\theta}_l>$	BU_WTHL	RES RELA	z,t	sponge layer relaxation
2way nesting of $<\tilde{w}\tilde{\theta}_l>$	BU_WTHL	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_WTHL	RES MISC	z,t	ref. pressure term, curvature term, microphysics, radiation, ...
residual of budget of $<\tilde{w}\tilde{\theta}_l>$	BU_WTHL	RES RESI	z,t	must be zero

field	notation in diac. file	processus name	dim.	comments
$-\frac{\partial}{\partial t} < \overline{w'\theta_l'} >$ (neglected in turb. scheme)	BU_WTHL	NSG TEND	z,t	(neglected) opposite of tendency of $< \overline{w'\theta_l'} >$
$- < w > \frac{\partial}{\partial z} < \overline{w'\theta_l'} >$	BU_WTHL	NSG ADVM	z,t	(neglected) advection by mean flow
$- < \tilde{u_\alpha} \frac{\partial}{\partial x_\alpha} \overline{w'\theta_l'} >$	BU_WTHL	NSG ADVR	z,t	(neglected) advection by resolved flow
terms due to $\overline{w}$ gradients	BU_WTHL	NSG DPGW	z,t	(neglected) dyn. prod. terms
terms due to hor. $\overline{\theta_l}$ gradients	BU_WTHL	NSG DPGT	z,t	other (neglected) dyn. prod. terms

C.13.3 Budget of total (liquid) temperature variance

All terms of the equation of $\frac{\partial}{\partial t}(< \tilde{\theta_l}^2 > + < \overline{\theta_l'^2} >)$ are computed and stored in the diachronic group BU_THL2. All comments made for the total Tke equation are valid here.

$$
\frac{\partial}{\partial t} < \overline{\theta_l'^2} > = \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < \overline{\theta_l'^2} >}^{ADVM} \overbrace{- < \tilde{u_\alpha} \frac{\partial}{\partial x_\alpha} \overline{\theta_l'^2} >}^{ADVR} \overbrace{-2 < \overline{u_\alpha'\theta_l'} > \frac{\partial < \theta_l >}{\partial x_\alpha}}^{DPM}
$$

$$
\underbrace{-2 < \overline{u_\alpha'\theta_l'} \frac{\partial \tilde{\theta_l}}{\partial x_\alpha} >}_{DPR} \underbrace{- < \frac{\partial}{\partial x_\alpha} \overline{u_\alpha'\theta_l'^2} >}_{TR} \underbrace{- \epsilon_\theta}_{DISS}
$$

$$
\frac{\partial}{\partial t} < \tilde{\theta_l}^2 > = \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < \tilde{\theta_l}^2 >}^{ADV} \overbrace{-2 < \tilde{u_\alpha}\tilde{\theta_l} > \frac{\partial < \theta_l >}{\partial x_\alpha}}^{DP}
$$

$$
\underbrace{-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{\theta_l}^2 >}_{TR} \underbrace{-2 < \tilde{\theta}_l \frac{\partial}{\partial x_\alpha} \overline{u_\alpha'\theta_l'} >}_{SBGT} + ...
$$

field	notation in diac. file	processus name	dim.	comments
$-2 < \overline{w'\theta_l'} > \frac{\partial}{\partial z} < \theta_l >$	BU_WTHL	SBG DP M	z,t	dyn. prod. by mean gradient
$< -2\overline{w'\theta_l'} \frac{\partial}{\partial z} \tilde{\theta}_l >$	BU_WTHL	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$- < \frac{\partial}{\partial z} \overline{w'\theta_l'^2} >$	BU_WTHL	SBG TR	z,t	subgrid turbulent transport
$- < \epsilon_\theta >$	BU_WTHL	SBG DISS	z,t	dissipation
residual of budget of $< \overline{w'\theta_l'} >$	BU_WTHL	SBG RESI	z,t	must be small

field	notation in diac. file	processus name	dim.	comments
$-\frac{\partial}{\partial t} < \tilde{\theta_l}^2 >$	BU_WTHL	RES TEND	z,t	(opposite of) tendency of $< \tilde{\theta_l}^2 >$
$- < w > \frac{\partial}{\partial z} < \tilde{\theta_l}^2 >$	BU_WTHL	RES ADV	z,t	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z} < \tilde{\theta_l}^2 >$	BU_WTHL	RES FORC	z,t	advection by large-scale W forcing
$- < 2\tilde{w}\tilde{\theta_l} > \frac{\partial}{\partial z} < \theta_l >$	BU_WTHL	RES DP	z,t	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{\theta_l}^2 >$	BU_WTHL	RES TR	z,t	resolved transport of resolved variance
$- < 2\tilde{\theta_l}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha\theta'_l} >$	BU_WTHL	RES SBGT	z,t	sink due to subgrid turbulence
numerical diffusion of $< \tilde{\theta_l}^2 >$	BU_WTHL	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $< \tilde{\theta_l}^2 >$	BU_WTHL	RES RELA	z,t	sponge layer relaxation
2way nesting of $< \tilde{\theta_l}^2 >$	BU_WTHL	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_WTHL	RES MISC	z,t	ref. pressure term, radiation, microphysics, ...
residual of budget of $< \tilde{\theta_l}^2 >$	BU_WTHL	RES RESI	z,t	must be zero
$-\frac{\partial}{\partial t} < \overline{\theta_l'^2} >$ (neglected in turb. scheme)	BU_WTHL	NSG TEND	z,t	(neglected) opposite of tendency of $< \overline{\theta_l'^2} >$
$- < w > \frac{\partial}{\partial z} < \overline{\theta_l'^2} >$	BU_WTHL	NSG ADVM	z,t	(neglected) advection by mean flow
$- < \tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{\theta_l'^2} >$	BU_WTHL	NSG ADVR	z,t	(neglected) advection by resolved flow

C.13.4 Budget of total water flux

All terms of the equation of $\frac{\partial}{\partial t}(<\tilde{w}\tilde{r}_t> + <\overline{w'r'_t}>)$ are computed and stored in the diachronic group BU_WRT. All comments made for the total Tke equation are valid here.

$$\frac{\partial}{\partial t}<\overline{w'r'_t}>= \overbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\overline{w'r'_t}>}^{ADVM}\ \overbrace{-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{w'r'_t}>}^{ADVR}\quad \overbrace{-<\overline{u'_\alpha w'}>\frac{\partial<r_t>}{\partial x_\alpha}-<\overline{u'_\alpha r'_t}>\frac{\partial<w>}{\partial x_\alpha}}^{DPM}$$
$$\underbrace{-<\overline{u'_\alpha w'}\frac{\partial\tilde{r}_t}{\partial x_\alpha}>-<\overline{u'_\alpha r'_t}\frac{\partial\tilde{w}}{\partial x_\alpha}>}_{DPR}+\underbrace{<\beta\overline{r'_t\theta'_v}>}_{TP}\ \underbrace{-\frac{1}{<\rho>}<\overline{r'_t\frac{\partial p'}{\partial z}}>}_{PRES}\quad\underbrace{-<\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'r'_t}>}_{TR}$$

$$\frac{\partial}{\partial t}<\tilde{w}\tilde{r}_t>= \overbrace{-<u_\alpha>\frac{\partial}{\partial x_\alpha}<\tilde{w}\tilde{r}_t>}^{ADV}\ \overbrace{-\frac{1}{<\rho>}<\tilde{r}_t\frac{\partial\tilde{p}}{\partial x_\alpha}>}^{PRES}\quad\overbrace{-<\tilde{u_\alpha}\tilde{w}>\frac{\partial<r_t>}{\partial x_\alpha}-<\tilde{u_\alpha}\tilde{r}_t>\frac{\partial<w>}{\partial x_\alpha}}^{DP}$$
$$+\underbrace{<\beta\tilde{r}_t\tilde{\theta}_v>}_{TP}\quad\underbrace{-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{w}\tilde{r}_t>}_{TR}\quad\underbrace{-<\tilde{w}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha r'_t}>-<\tilde{r}_t\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'}>}_{SBGT}+...$$

field	notation in diac. file	processus name	dim.	comments
$-<\overline{w'^2}>\frac{\partial}{\partial z}<r_t>$	BU_WRT	SBG DP M	z,t	dyn. prod. by mean gradient
$<-\overline{w'^2}\frac{\partial}{\partial z}\tilde{r}_t>$	BU_WRT	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$-<\frac{\partial}{\partial z}\overline{w'^2r'_t}>$	BU_WRT	SBG TR	z,t	subgrid turbulent transport
$-\frac{1}{<\rho>}<\overline{r'_t\frac{\partial p'}{\partial z}}>$	BU_WRT	SBG PRES	z,t	subgrid pressure-correlation term
$<\beta\overline{r'_t\theta'_v}>$	BU_WRT	SBG TP	z,t	thermal production
residual of budget of $<\overline{w'r'_t}>$	BU_WRT	SBG RESI	z,t	must be small
$-\frac{\partial}{\partial t}<\tilde{w}\tilde{r}_t>$	BU_WRT	RES TEND	z,t	(opposite of) tendency of $<\tilde{w}\tilde{r}_t>$
$-<w>\frac{\partial}{\partial z}<\tilde{w}\tilde{r}_t>$	BU_WRT	RES ADV	z,t	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z}<\tilde{w}\tilde{r}_t>$	BU_WRT	RES FORC	z,t	advection by large-scale W forcing
$-<\tilde{w}^2>\frac{\partial}{\partial z}<r_t>$ $-<\tilde{w}\tilde{r}_t>\frac{\partial}{\partial z}<w>$	BU_WRT	RES DP	z,t	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{w}\tilde{r}_t>$	BU_WRT	RES TR	z,t	transport of resolved flux by itself
$-\frac{1}{<\rho>}<\tilde{r}_t\frac{\partial\tilde{p}}{\partial z}>$	BU_WRT	RES PRES	z,t	pressure-correlations

field	notation in diac. file	processus name	dim.	comments
$<\beta\tilde{r_t}\tilde{\theta_v}>$	BU_WRT	RES TP	z,t	thermal production
$-<\tilde{w}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha r'_t}>$ $-<\tilde{r}_t\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'}>$	BU_WRT	RES SBGT	z,t	sink due to subgrid turbulence
Coriolis terms	BU_WRT	RES CORI	z,t	
numerical diffusion of $<\tilde{w}\tilde{r_t}>$	BU_WRT	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $<\tilde{w}\tilde{r_t}>$	BU_WRT	RES RELA	z,t	sponge layer relaxation
2way nesting of $<\tilde{w}\tilde{r_t}>$	BU_WRT	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_WRT	RES MISC	z,t	ref. pressure term, curvature term, radiation, microphysics, ...
residual of budget of $<\tilde{w}\tilde{r_t}>$	BU_WRT	RES RESI	z,t	must be zero
$-\frac{\partial}{\partial t}<\overline{w'r'_t}>$ (neglected in turb. scheme)	BU_WRT	NSG TEND	z,t	(neglected) opposite of tendency of $<\overline{w'r'_t}>$
$-<w>\frac{\partial}{\partial z}<\overline{w'r'_t}>$	BU_WRT	NSG ADVM	z,t	(neglected) advection by mean flow
$-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{w'r'_t}>$	BU_WRT	NSG ADVR	z,t	(neglected) advection by resolved flow
terms due to $\overline{w}$ gradients	BU_WRT	NSG DPGW	z,t	(neglected) dyn. prod. terms
terms due to hor. $\overline{r_t}$ gradients	BU_WRT	NSG DPGT	z,t	other (neglected) dyn. prod. terms

C.13.5 Budget of liquid temperature - total water covariance

All terms of the equation of $\frac{\partial}{\partial t}(<\tilde{\theta}_l \tilde{r}_t> + <\overline{\theta'_l r'_t}>)$ are computed and stored in the diachronic group BU_THLR. All comments made for the total Tke equation are valid here.

$$\frac{\partial}{\partial t} <\overline{\theta'_l r'_t}> = \overbrace{-<u_\alpha> \frac{\partial}{\partial x_\alpha} <\overline{\theta'_l r'_t}>}^{ADVM} \overbrace{-<\tilde{u_\alpha} \frac{\partial}{\partial x_\alpha} \overline{\theta'_l r'_t}>}^{ADVR} \overbrace{-<\overline{u'_\alpha \theta'_l}> \frac{\partial <r_t>}{\partial x_\alpha} - <\overline{u'_\alpha r'_t}> \frac{\partial <\theta_l>}{\partial x_\alpha}}^{DPM}$$
$$\underbrace{-<\overline{u'_\alpha \theta'_l} \frac{\partial \tilde{r}_t}{\partial x_\alpha}> - <\overline{u'_\alpha r'_t} \frac{\partial \tilde{\theta}_l}{\partial x_\alpha}>}_{DPR} \underbrace{-<\frac{\partial}{\partial x_\alpha} \overline{u'_\alpha \theta'_l r'_t}>}_{TR} \underbrace{-\epsilon_{\theta r}}_{DISS}$$

$$\frac{\partial}{\partial t} <\tilde{\theta}_l \tilde{r}_t> = \overbrace{-<u_\alpha> \frac{\partial}{\partial x_\alpha} <\tilde{\theta}_l \tilde{r}_t>}^{ADV} \overbrace{-<\tilde{u_\alpha} \tilde{\theta}_l> \frac{\partial <r_t>}{\partial x_\alpha} - <\tilde{u_\alpha} \tilde{r}_t> \frac{\partial <\theta_l>}{\partial x_\alpha}}^{DP}$$
$$\underbrace{-\frac{\partial}{\partial x_\alpha} <\tilde{u_\alpha} \tilde{\theta}_l \tilde{r}_t>}_{TR} \underbrace{-<\tilde{\theta}_l \frac{\partial}{\partial x_\alpha} \overline{u'_\alpha r'_t}> - <\tilde{r}_t \frac{\partial}{\partial x_\alpha} \overline{u'_\alpha \theta'_l}>}_{SBGT} + ...$$

field	notation in diac. file	processus name	dim.	comments
$-<\overline{w'\theta'_l}> \frac{\partial}{\partial z} <r_t>$ $-<\overline{w'r'_t}> \frac{\partial}{\partial z} <\theta_l>$	BU_THLR	SBG DP M	z,t	dyn. prod. by mean gradient
$<-\overline{u'_\alpha \theta'_l} \frac{\partial}{\partial x_\alpha} \tilde{r}_t>$ $<-\overline{u'_\alpha r'_t} \frac{\partial}{\partial x_\alpha} \tilde{\theta}_l>$	BU_THLR	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$-<\frac{\partial}{\partial z} \overline{w'\theta'_l r'_t}>$	BU_THLR	SBG TR	z,t	subgrid turbulent transport
$-<\epsilon_{\theta r}>$	BU_THLR	SBG DISS	z,t	dissipation
residual of budget of $<\overline{w'\theta'_l r'_t}>$	BU_THLR	SBG RESI	z,t	must be small
$-\frac{\partial}{\partial t} <\tilde{\theta}_l \tilde{r}_t>$	BU_THLR	RES TEND	z,t	(opposite of) tendency of $<\tilde{\theta}_l \tilde{r}_t>$
$-<w> \frac{\partial}{\partial z} <\tilde{\theta}_l \tilde{r}_t>$	BU_THLR	RES ADV	z,t	advection by mean flow
$-W_{forc} \frac{\partial}{\partial z} <\tilde{\theta}_l \tilde{r}_t>$	BU_THLR	RES FORC	z,t	advection by large-scale W forcing
$-<\tilde{w}\tilde{\theta}_l> \frac{\partial}{\partial z} <r_t>$ $-<\tilde{w}\tilde{r}_t> \frac{\partial}{\partial z} <\theta_l>$	BU_THLR	RES DP	z,t	dyn. prod. (by mean gradients)

field	notation in diac. file	processus name	dim.	comments
$-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES TR	z,t	resolved transport of resolved flux
$- < \tilde{\theta_l}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha r'_t} >$ $- < \tilde{r_t}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha \theta'_l} >$	BU_THLR	RES SBGT	z,t	sink due to subgrid turbulence
numerical diffusion of $< \tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $< \tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES RELA	z,t	sponge layer relaxation
2way nesting of $< \tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_THLR	RES MISC	z,t	ref. pressure term, radiation, microphysics, ...
residual of budget of $< \tilde{\theta_l}\tilde{r_t} >$	BU_THLR	RES RESI	z,t	must be zero
$-\frac{\partial}{\partial t} < \overline{\theta'_l r'_t} >$ (neglected in turb. scheme)	BU_THLR	NSG TEND	z,t	(neglected) opposite of tendency of $< \overline{w' r'_t} >$
$- < w > \frac{\partial}{\partial z} < \overline{\theta'_l r'_t} >$	BU_THLR	NSG ADVM	z,t	(neglected) advection by mean flow
$- < \tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{\theta'_l r'_t} >$	BU_THLR	NSG ADVR	z,t	(neglected) advection by resolved flow

C.13.6 Budget of total water variance

All terms of the equation of $\frac{\partial}{\partial t}(< \tilde{r_t}^2 > + < \overline{r'^2_t} >)$ are computed and stored in the diachronic group BU_RT2. All comments made for the total Tke equation are valid here.

$$
\frac{\partial}{\partial t} < \overline{r'^2_t} > = \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < \overline{r'^2_t} >}^{ADVM} \overbrace{- < \tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{r'^2_t} >}^{ADVR} \overbrace{-2 < \overline{u'_\alpha r'_t} > \frac{\partial < r_t >}{\partial x_\alpha}}^{DPM}
$$

$$
\underbrace{-2 < \overline{u'_\alpha r'_t}\frac{\partial \tilde{r_t}}{\partial x_\alpha} >}_{DPR} \underbrace{- < \frac{\partial}{\partial x_\alpha}\overline{u'_\alpha r'^2_t} >}_{TR} \underbrace{-\epsilon_r}_{DISS}
$$

$$
\frac{\partial}{\partial t} < \tilde{r_t}^2 > = \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < \tilde{r_t}^2 >}^{ADV} \overbrace{-2 < \tilde{u_\alpha}\tilde{r_t} > \frac{\partial < r_t >}{\partial x_\alpha}}^{DP}
$$

$$
\underbrace{-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{r_t}^2 >}_{TR} \underbrace{-2 < \tilde{r_t}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha r'_t} >}_{SBGT} + ...
$$

field	notation in diac. file	processus name	dim.	comments
$-2 < \overline{w'r'_t} > \frac{\partial}{\partial z} < r_t >$	BU_RT2	SBG DP M	z,t	dyn. prod. by mean gradient
$< -2\overline{w'r'_t}\frac{\partial}{\partial z}\tilde{r}_t >$	BU_RT2	SBG DP R	z,t	dyn. prod. by resolved fluctuations
$- < \frac{\partial}{\partial z}\overline{w'r'^2_t} >$	BU_RT2	SBG TR	z,t	subgrid turbulent transport
$- < \epsilon_r >$	BU_RT2	SBG DISS	z,t	dissipation
$-\frac{\partial}{\partial t} < \tilde{r_t}^2 >$	BU_RT2	RES TEND	z,t	(opposite of) tendency of $< \tilde{r_t}^2 >$
$- < w > \frac{\partial}{\partial z} < \tilde{r_t}^2 >$	BU_RT2	RES ADV	z,t	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z} < \tilde{r_t}^2 >$	BU_RT2	RES FORC	z,t	advection by large-scale W forcing
$- < 2\tilde{w}\tilde{\theta_l} > \frac{\partial}{\partial z} < r_t >$	BU_RT2	RES DP	z,t	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{r_t}^2 >$	BU_RT2	RES TR	z,t	resolved transport of resolved variance
$- < 2\tilde{r}_t\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha r'_t} >$	BU_RT2	RES SBGT	z,t	sink due to subgrid turbulence
numerical diffusion of $< \tilde{r_t}^2 >$	BU_RT2	RES NUMD	z,t	numerical diffusion against $2\Delta x$
relaxation of $< \tilde{r_t}^2 >$	BU_RT2	RES RELA	z,t	sponge layer relaxation
2way nesting of $< \tilde{r_t}^2 >$	BU_RT2	RES NEST	z,t	average from smaller nested models
miscellaneous	BU_RT2	RES MISC	z,t	ref. pressure term, radiation, microphysics, ...
residual of budget of $< \tilde{r_t}^2 >$	BU_RT2	RES RESI	z,t	must be zero
$-\frac{\partial}{\partial t} < \overline{r'^2_t} >$ (neglected in turb. scheme)	BU_RT2	NSG TEND	z,t	(neglected) opposite of tendency of $< \overline{r'^2_t} >$
$- < w > \frac{\partial}{\partial z} < \overline{r'^2_t} >$	BU_RT2	NSG ADVM	z,t	(neglected) advection by mean flow
$- < \tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{r'^2_t} >$	BU_RT2	NSG ADVR	z,t	(neglected) advection by resolved flow

C.13.7 Budget of total scalar flux

All terms of the equation of $\frac{\partial}{\partial t}(<\tilde{w}\tilde{s_v}> + <\overline{w's'_v}>)$ are computed and stored in the diachronic group BU_WSV. All comments made for the total Tke equation are valid here.

$$
\begin{aligned}
\frac{\partial}{\partial t} <\overline{w's'_v}> = \; & \overbrace{-<u_\alpha> \frac{\partial}{\partial x_\alpha} <\overline{w's'_v}>}^{ADVM} \overbrace{-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{w's'_v}>}^{ADVR} \overbrace{-<\overline{u'_\alpha w'}\frac{\partial \tilde{s_v}}{\partial x_\alpha}> - <\overline{u'_\alpha s'_v}\frac{\partial \tilde{w}}{\partial x_\alpha}>}^{DPR} \\
& \underbrace{-<\overline{u'_\alpha w'}> \frac{\partial <s_v>}{\partial x_\alpha} - <\overline{u'_\alpha s'_v}> \frac{\partial <w>}{\partial x_\alpha}}_{DPM} + \underbrace{<\beta \overline{s'_v \theta'_v}>}_{TP} \underbrace{-\frac{1}{<\rho>} <\overline{s'_v\frac{\partial p'}{\partial z}}>}_{PRES} \\
& \underbrace{-<\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w' s'_v}>}_{TR}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial}{\partial t} <\tilde{w}\tilde{s_v}> = \; & \overbrace{-<u_\alpha> \frac{\partial}{\partial x_\alpha} <\tilde{w}\tilde{s_v}>}^{ADV} \overbrace{-\frac{1}{<\rho>} <\tilde{s_v}\frac{\partial \tilde{p}}{\partial x_\alpha}>}^{PRES} + \overbrace{<\beta \tilde{s_v}\tilde{\theta_v}>}^{TP} \\
& \underbrace{-<\tilde{w}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha s'_v}> - <\tilde{s_v}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'}>}_{SBGT} \underbrace{-\frac{\partial}{\partial x_\alpha} <\tilde{u_\alpha}\tilde{w}\tilde{s_v}>}_{TR} \\
& \underbrace{-<\tilde{u_\alpha}\tilde{w}> \frac{\partial <s_v>}{\partial x_\alpha} - <\tilde{u_\alpha}\tilde{s_v}> \frac{\partial <w>}{\partial x_\alpha}}_{DP} + \ldots
\end{aligned}
$$

field	notation in diac. file	processus name	dim.	comments
$-<\overline{w'^2}> \frac{\partial}{\partial z} <s_v>$	BU_WSV	SBG DP M	z,t,n	dyn. prod. by mean gradient
$<-\overline{w'^2}\frac{\partial}{\partial z}\tilde{s_v}>$	BU_WSV	SBG DP R	z,t,n	dyn. prod. by resolved fluctuations
$-<\frac{\partial}{\partial z}\overline{w'^2 s'_v}>$	BU_WSV	SBG TR	z,t,n	subgrid turbulent transport
$-\frac{1}{<\rho>} <\overline{s'_v\frac{\partial p'}{\partial z}}>$	BU_WSV	SBG PRES	z,t,n	subgrid pressure-correlation term
$<\beta\overline{s'_v\theta'_v}>$	BU_WSV	SBG TP	z,t,n	thermal production
residual of budget of $<\overline{w's'_v}>$	BU_WSV	SBG RESI	z,t,n	must be small
$-\frac{\partial}{\partial t} <\tilde{w}\tilde{s_v}>$	BU_WSV	RES TEND	z,t,n	(opposite of) tendency of $<\tilde{w}\tilde{s_v}>$
$-<w> \frac{\partial}{\partial z} <\tilde{w}\tilde{s_v}>$	BU_WSV	RES ADV	z,t,n	advection by mean flow
$-W_{forc}\frac{\partial}{\partial z} <\tilde{w}\tilde{s_v}>$	BU_WSV	RES FORC	z,t,n	advection by large-scale W forcing

field	notation in diac. file	processus name	dim.	comments
$-<\tilde{w}^2>\frac{\partial}{\partial z}<s_v>$ $-<\tilde{w}\tilde{s_v}>\frac{\partial}{\partial z}<w>$	BU_WSV	RES DP	z,t,n	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha}<\tilde{u_\alpha}\tilde{w}\tilde{s_v}>$	BU_WSV	RES TR	z,t,n	transport of resolved flux by itself
$-\frac{1}{<\rho>}<\tilde{s_v}\frac{\partial \tilde{p}}{\partial z}>$	BU_WSV	RES PRES	z,t,n	pressure-correlations
$<\beta\tilde{s_v}\tilde{\theta_v}>$	BU_WSV	RES TP	z,t,n	thermal production
$-<\tilde{w}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha s'_v}>$ $-<\tilde{s_v}\frac{\partial}{\partial x_\alpha}\overline{u'_\alpha w'}>$	BU_WSV	RES SBGT	z,t,n	sink due to subgrid turbulence
Coriolis terms	BU_WSV	RES CORI	z,t,n	
numerical diffusion of $<\tilde{w}\tilde{s_v}>$	BU_WSV	RES NUMD	z,t,n	numerical diffusion against $2\Delta x$
relaxation of $<\tilde{w}\tilde{s_v}>$	BU_WSV	RES RELA	z,t,n	sponge layer relaxation
2way nesting of $<\tilde{w}\tilde{s_v}>$	BU_WSV	RES NEST	z,t,n	average from smaller nested models
miscellaneous	BU_WSV	RES MISC	z,t,n	curvature term, chemistry, ...
residual of budget of $<\tilde{w}\tilde{s_v}>$	BU_WSV	RES RESI	z,t,n	must be zero
$-\frac{\partial}{\partial t}<\overline{w's'_v}>$ (neglected in turb. scheme)	BU_WSV	NSG TEND	z,t,n	(neglected) opposite of tendency of $<\overline{w's'_v}>$
$-<w>\frac{\partial}{\partial z}<\overline{w's'_v}>$	BU_WSV	NSG ADVM	z,t,n	(neglected) advection by mean flow
$-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{w's'_v}>$	BU_WSV	NSG ADVR	z,t,n	(neglected) advection by resolved flow
terms due to $\overline{w}$ gradients	BU_WSV	NSG DPGW	z,t,n	(neglected) dyn. prod. terms
terms due to hor. $\overline{s_v}$ gradients	BU_WSV	NSG DPGT	z,t,n	other (neglected) dyn. prod. terms

C.13.8 Budget of total scalar variance

All terms of the equation of $\frac{\partial}{\partial t}(< \tilde{s_v}^2 > + < \overline{s_v'^2} >)$ are computed and stored in the diachronic group BU_SV2. All comments made for the total Tke equation are valid here.

$$\frac{\partial}{\partial t} < \overline{s_v'^2} >= \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < \overline{s_v'^2} >}^{ADVM} \quad \overbrace{- < \tilde{u_\alpha} \frac{\partial}{\partial x_\alpha} \overline{s_v'^2} >}^{ADVR} \quad \overbrace{-2 < \overline{u_\alpha' s_v'} > \frac{\partial < s_v >}{\partial x_\alpha}}^{DPM}$$
$$\underbrace{-2 < \overline{u_\alpha' s_v'} \frac{\partial \tilde{s_v}}{\partial x_\alpha} >}_{DPR} \quad \underbrace{- < \frac{\partial}{\partial x_\alpha} \overline{u_\alpha' s_v'^2} >}_{TR} \quad \underbrace{- \epsilon_{s_v}}_{DISS}$$

$$\frac{\partial}{\partial t} < \tilde{s_v}^2 >= \overbrace{- < u_\alpha > \frac{\partial}{\partial x_\alpha} < \tilde{s_v}^2 >}^{ADV} \quad \overbrace{-2 < \tilde{u_\alpha}\tilde{s_v} > \frac{\partial < s_v >}{\partial x_\alpha}}^{DP}$$
$$\underbrace{-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{s_v}^2 >}_{TR} \quad \underbrace{-2 < \tilde{s_v} \frac{\partial}{\partial x_\alpha}\overline{u_\alpha' s_v'} >}_{SBGT} \quad +...$$

field	notation in diac. file	processus name	dim.	comments
$-2 < \overline{w' s_v'} > \frac{\partial}{\partial z} < s_v >$	BU_SV2	SBG DP M	z,t,n	dyn. prod. by mean gradient
$< -2\overline{w' s_v'} \frac{\partial}{\partial z} \tilde{s_v} >$	BU_SV2	SBG DP R	z,t,n	dyn. prod. by resolved fluctuations
$- < \frac{\partial}{\partial z} \overline{w' s_v'^2} >$	BU_SV2	SBG TR	z,t,n	subgrid turbulent transport
$- < \epsilon_{s_v} >$	BU_SV2	SBG DISS	z,t,n	dissipation
residual of budget of $< \overline{w' s_v'} >$	BU_SV2	SBG RESI	z,t,n	must be small
$-\frac{\partial}{\partial t} < \tilde{s_v}^2 >$	BU_SV2	RES TEND	z,t,n	(opposite of) tendency of $< \tilde{s_v}^2 >$
$- < w > \frac{\partial}{\partial z} < \tilde{s_v}^2 >$	BU_SV2	RES ADV	z,t,n	advection by mean flow
$-W_{forc} \frac{\partial}{\partial z} < \tilde{s_v}^2 >$	BU_SV2	RES FORC	z,t,n	advection by large-scale W forcing
$- < 2\tilde{w}\tilde{\theta_l} > \frac{\partial}{\partial z} < s_v >$	BU_SV2	RES DP	z,t,n	dyn. prod. (by mean gradients)
$-\frac{\partial}{\partial x_\alpha} < \tilde{u_\alpha}\tilde{s_v}^2 >$	BU_SV2	RES TR	z,t,n	resolved transport of resolved variance
$- < 2\tilde{s_v} \frac{\partial}{\partial x_\alpha} \overline{u_\alpha' s_v'} >$	BU_SV2	RES SBGT	z,t,n	sink due to subgrid turbulence
numerical diffusion of $< \tilde{s_v}^2 >$	BU_SV2	RES NUMD	z,t,n	numerical diffusion against $2\Delta x$
relaxation of $< \tilde{s_v}^2 >$	BU_SV2	RES RELA	z,t,n	sponge layer relaxation

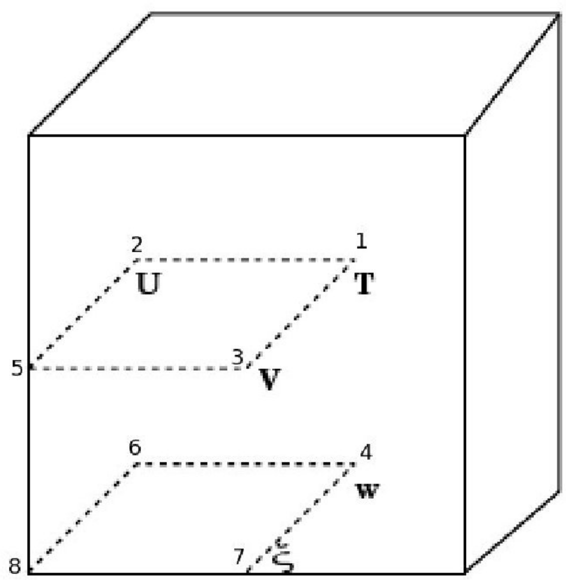
field	notation in diac. file	processus name	dim.	comments
2way nesting of $<\tilde{s_v}^2>$	BU_SV2	RES NEST	z,t,n	average from smaller nested models
miscellaneous	BU_SV2	RES MISC	z,t,n	chemistry, ...
residual of budget of $<\tilde{s_v}^2>$	BU_SV2	RES RESI	z,t,n	must be zero
$-\frac{\partial}{\partial t}<\overline{s_v'^2}>$ (neglected in turb. scheme)	BU_SV2	NSG TEND	z,t,n	(neglected) opposite of tendency of $<\overline{s_v'^2}>$
$-<w>\frac{\partial}{\partial z}<\overline{s_v'^2}>$	BU_SV2	NSG ADVM	z,t,n	(neglected) advection by mean flow
$-<\tilde{u_\alpha}\frac{\partial}{\partial x_\alpha}\overline{s_v'^2}>$	BU_SV2	NSG ADVR	z,t,n	(neglected) advection by resolved flow

Appendix D

MESONH grid

MESO-NH use a C-grid in the Arakawa convention, both on the horizontal and on the vertical. This grid is shown on the following figure :

- 1 : mass points
- 2 : u points
- 3 : v points
- 4 : w points
- 5 : vertical vorticity point
- 6 : vorticity components along y
- 7 : vorticity components along x



Index